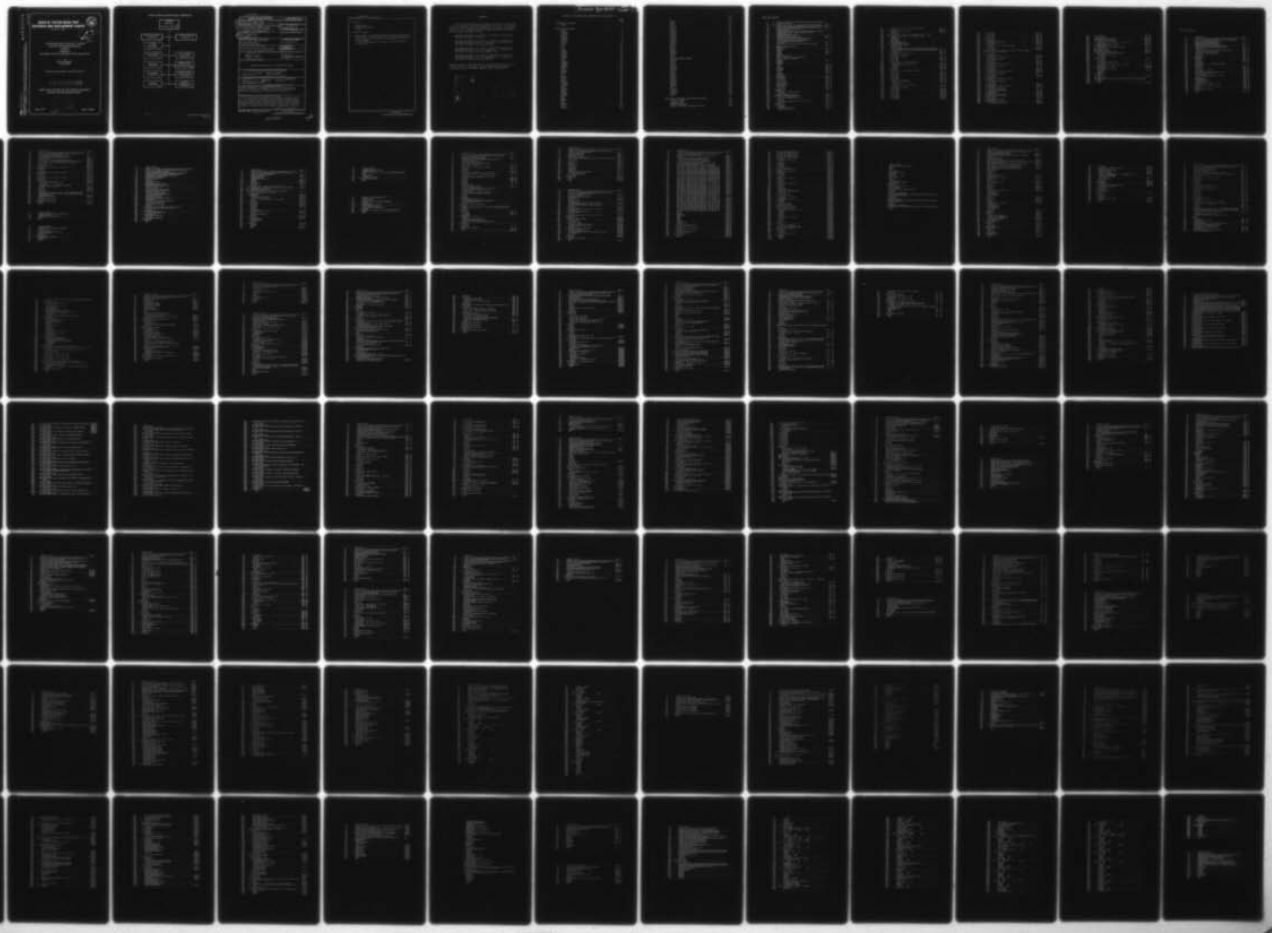
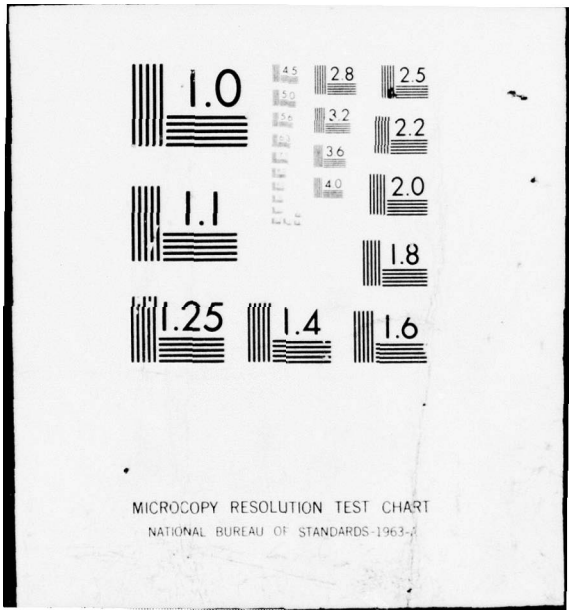
AD-A043 924 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 9/2
MAINTENANCE MANUAL FOR AUDIT, A SYSTEM FOR ANALYZING SESCOMP SO--ETC(U)
AUG 77 R J WYBRANIEC, R REGEN

UNCLASSIFIED DTNSRDC-77-0075-VOL-3 NL

1 OF 2
AD
A043924




1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

Report 77-0075

DAVID W. TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CENTER

Bethesda, Md. 20084

12
NW

MAINTENANCE MANUAL FOR AUDIT, A SYSTEM FOR ANALYZING SESCOMP SOFTWARE VOLUME 3 APPENDIX C LISTINGS OF THE AUDIT SOFTWARE FOR THE UNIVAC 1108

by

Robert J. Wybraniec
Richard Regen

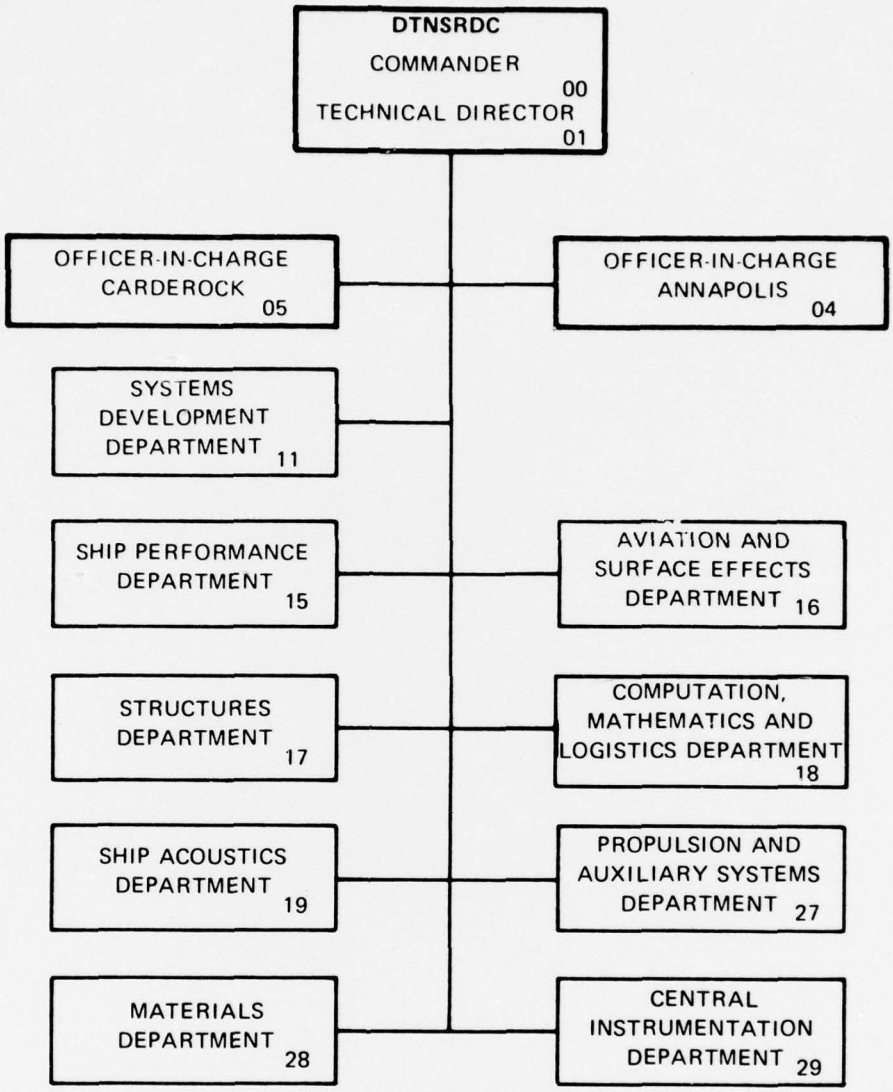
COMPUTATION, MATHEMATICS, AND LOGISTICS DEPARTMENT
RESEARCH AND DEVELOPMENT REPORT

August 1977

Report 77-0075

MAINTENANCE MANUAL FOR AUDIT, A SYSTEM FOR ANALYZING SESCOMP SOFTWA...
VOLUME 3 – APPENDIX C – LISTINGS OF THE AUDIT SOFTWARE FOR THE UNIVAC 1108

MAJOR DTNSRDC ORGANIZATIONAL COMPONENTS



GPO 905-678

NDW-DTNSRDC 3960/43b (Rev. 11-75)
GPO 928-108

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: DTNSRDC-77-0075-Vol-3	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): MAINTENANCE MANUAL FOR AUDIT, A SYSTEM FOR ANALYZING SESCOMP SOFTWARE, VOLUME 3: APPENDIX C - LISTINGS OF THE AUDIT SOFTWARE FOR THE UNIVAC 1108.	5. TYPE OF REPORT & PERIOD COVERED: Final rept. / 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Robert J. Wybraniec, Richard Regen	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: David W. Taylor Naval Ship Research and Development Center, Bethesda, Maryland 20084	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: (See reverse side)
11. CONTROLLING OFFICE NAME AND ADDRESS: Navy Surface Effect Ships Project (PMS 304), P.O. Box 34401 - Bethesda, Maryland 20084	12. REPORT DATE: August 1977 / 13. NUMBER OF PAGES: 161
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): 160p.	15. SECURITY CLASS. (of this report): UNCLASSIFIED / 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

APPROVED FOR PUBLIC RELEASE: DISTRIBUTION UNLIMITED

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report): SSH15, S0308001

18. SUPPLEMENTARY NOTES: SSH15001, S0308001

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

SESCOMP, SESCOMPSPEC's, Software Verification, Software Engineering, Reliability, Graph Theory, FORTRAN Software, Modules, Flow Analysis, Variable Precision Execution, Parser, Roll Call, Portability

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The AUDIT documentation provides the maintenance programmer personnel with the information to effectively maintain and use the AUDIT software. The AUDIT software examines FORTRAN computer programs or modules developed under the SESCOMP system for compliance with certain prescribed standards (SESCOMPSPEC's) and produces reports detailing the deviations from those standards. The AUDIT software also examines a program unit to detect and

(Continued on reverse side)

next page

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102-LF-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

387682

4/B

(Block 10)

63534N, 19588,
SSH15001 and S0308001,
11837001

(Block 20 continued)

report improper use of undefined variables along the program unit's possible paths. In addition, AUDIT has an option which enables the user to test the effect of changes in word length on the output of computer programs.

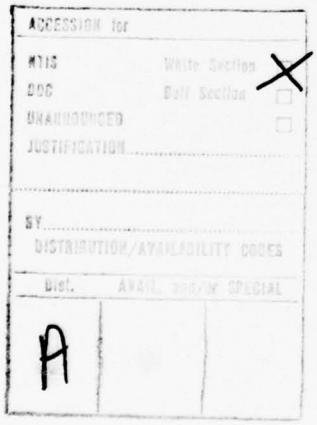
This report contains the listings of the AUDIT software for the UNIVAC 1108.

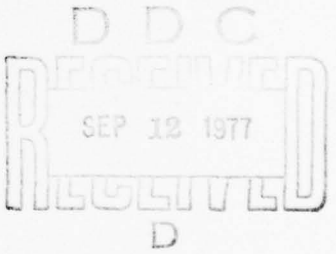
FOREWORD

The use and maintenance of AUDIT, a software system for analyzing SESCOMP contractor-supplied software, is documented as a set of four separately bound David W. Taylor Naval Ship Research and Development Center volumes sharing the common report number--DTNSRDC 77-0075:

- Maintenance Manual for AUDIT, a System for Analyzing SESCOMP Software, Volume 1
- Maintenance Manual for AUDIT, a System for Analyzing SESCOMP Software, Volume 2; Appendix B - Listings of the AUDIT Software for the CDC 6000
- Maintenance Manual for AUDIT, a System for Analyzing SESCOMP Software, Volume 3; Appendix C - Listings of the AUDIT Software for the UNIVAC 1108
- Maintenance Manual for AUDIT, a System for Analyzing SESCOMP Software, Volume 4; Appendix D - Listings of the AUDIT Software for the IBM 360

Volume 1 describes AUDIT and the use and maintenance of the AUDIT software. The other three volumes offer software listings for the CDC 6000, UNIVAC 1108, and IBM 360.





INDEX TO PROGRAMS AND SUBPROGRAMS IN APPENDIX C

AUDIT Main Program

```
      C     PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE7,TAPE8,
      C    * TAPE9,TAPE4,TAPE99)
      C****IF UNIVAC 1108 - PLACE A 'C' IN COLUMN 1 OF PREVIOUS TWO CARDS      CH34   1
            COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),   RICH   2
           * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
           2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                   RICH   4
            DIMENSION IORD(15)
            COMMON/GLOBAL/NBLK,NREF,NSUBS,BLKTBL(200),EXTTBL(100),ISUBS(100)
            COMMON/LIST/NLIST,NINTFC,ISUBLT(3,200),INTFAC(500)
            COMMON/FLOW/IFL,IRP
            COMMON/STFUNC/NSTFNC,ISTFNC(10)
            COMMON/INPOUT/NCALL,IN,IOP                                         MAIN   7
            COMMON/LABELS/STATRA(2,200),NLABEL                                 MAIN   8
            COMMON/DOLOOP/ISTACK(4,50),NSTACK,ILOOP,IOVFL#
            COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
            INTEGER A,BLANK,STATRA,EXTTBL,BLKTBL
            DATA BLANK/1H /                                                    MAIN   15
            DATA IORD/29,30,31,32,25,28,19,20,21,22,23,24,26,27,35/
            IOP=8                                                              MAIN   18
            READ(5,12) MODE,IN,IFL,INTR
         12 FORMAT(11I,3I4)
            NCALL=0                                                            CH34   2
            NPROG=99
            NREF=0                                                             CH34   3
            NBLK=0                                                             CH34   4
            NSUBS=0                                                            CH34   5
            IRP=IFL-1
            READ(4) NLIST,NINTFC
            READ(4) ((ISUBLT(I,J),I=1,3),J=1,NLIST)
            READ(4) (INTFAC(I),I=1,NINTFC)
          4 ISTAT=0                                                            CH34   12
            IOVFL#=0
            MAINPR=0
            NLABEL=0
            NID=0                                                              CH34   13
            JJ=0                                                               CH34   14
            IBLKDT=0                                                           CH34   15
            NSTACK=0                                                           CH34   16
            NBLOCK=0                                                           CH34   17
            ILOOP=0                                                            CH34   18
            NB=0                                                               CH34   19
            IBLKST=0                                                           CH34   20
            NSTFNC=0
            #RITE(6,13)                                                       MAIN   38
         13 FORMAT(1H1)                                                        MAIN   39
            IFNCNM=BLANK                                                       MAIN   40
            DO 2 I=1,8                                                         MAIN   41
            DO 2 J=1,500                                                       MAIN   42
          2 IDTBL(I,J)=0                                                       MAIN   43
            DO 3 I=1,200                                                       MAIN   44
            STATRA(1,I)=0                                                      MAIN   45
          3 STATRA(2,I)=0                                                      MAIN   46
            DO 5 I=1,3                                                         MAIN   47
            INITID(I)=0                                                        MAIN   48
          5 LASTID(I)=0                                                        MAIN   49
        700 CONTINUE
            IF(NBLOCK .GT. 2500) GO TO 7000
            CALL BUILD                                                         MAIN   51
            #RITE(6,1000) (A(I),I=1,N)                                         MAIN   52
       1000 FORMAT(//6X,100A1,13(/6X,100A1))                                   MAIN   53
            IF(A(1) .EQ. IHC) GO TO 700                                        MAIN   54
            IF(NEXT(1) .EQ. BLANK) GO TO 700                                   MAIN   55
            JJ=JJ+1                                                            MAIN   56
            CALL CLASS                                                         MAIN   57
            IF(ITYP .GT. 18 .AND. ITYP .NE. 28) GO TO 1320                     MAIN   58
            CALL STATNO                                                        MAIN   59
       1320 JPTR=7                                                             MAIN   60
            IPREC=ISTAT                                                        MAIN   61
            ISTAT=ITYP                                                         MAIN   62
            IF(JJ .EQ. 1) GO TO 7                                              MAIN   63
            IF(IBLKDT .EQ. 0) GO TO 6                                          MAIN   64
```

```
      IF(ITYP .GE. 18 .AND. ITYP .LE. 27) GO TO 8                    MAIN  65
      GO TO 220                                                      MAIN  66
    6 IF(LTYP .EQ. 9) GO TO 120                                      MAIN  67
      GO TO 8                                                        MAIN  68
    7 IPREC=29
C*****IF CDC 6700 - PLACE A :C: IN COLUMN 1 OF NEXT  4 CARDS
      IF(ITYP .GE. 29 .AND. ITYP .LE. 32) GO TO 25
      MAINPR=1
      NXTID=4HMAIN
      CALL PROG
   25 NPROG=NPROG+1
      WRITE(IOP,900) NPROG
  900 FORMAT(1OH0FOR,I PRG,I3)
      IF(MAINPR .EQ. 0) GO TO 8
      WRITE(IOP,610)
      WRITE(IOP,620)
    8 GO TO (60,70,80,90,100,110,20,140,20,20,40,40,50,50,50,120,130,  MAIN  70
     1 2000,10,10,10,10,10,170,180,190,200,210,150,38,30,220,220,220,
     2 220,220),ITYP                                                 MAIN  72
   10 DO 15 I=1,11                                                   MAIN  73
      IF(IPREC .EQ. IORD(I)) GO TO 17                                MAIN  74
   15 CONTINUE                                                       MAIN  75
      CALL ERROR(2,KDM1,KDM2)
   17 CALL TYPE                                                      MAIN  77
      GO TO 500                                                      MAIN  78
   20 CALL SIMP                                                      MAIN  79
      GO TO 500                                                      MAIN  80
   30 IF(JJ .NE. 1) CALL ERROR(2,KDM1,KDM2)
      DO 35 I=1,20                                                   MAIN  82
      IF(NEXT(JPTR) .NE. IHF) GO TO 35                               MAIN  83
      JPTR=JPTR-1                                                    MAIN  84
      CALL SUB                                                       MAIN  85
      GO TO 500                                                      MAIN  86
   35 CONTINUE                                                       MAIN  87
   38 IF(JJ .NE. 1) CALL ERROR(2,KDM1,KDM2)
      CALL SUB                                                       MAIN  89
      GO TO 500                                                      MAIN  90
   40 CALL IO                                                        MAIN  91
      GO TO 500                                                      MAIN  92
   50 CALL AUXIO                                                     MAIN  93
      GO TO 500                                                      MAIN  94
   60 CALL INIT                                                      MAIN  95
      WRITE(6,66) (A(I),I=1,N)                                       MAIN  96
      IF(ITYP .NE. 35) GO TO 500                                     MAIN  97
      ISTAT=35                                                       MAIN  98
      DO 67 I=1,15
      IF(IPREC .EQ. IORD(I)) GO TO 500                               MAIN 100
   67 CONTINUE                                                       MAIN 101
      CALL ERROR(2,KDM1,KDM2)
      GO TO 500                                                      MAIN 103
   70 CALL ASSIGN                                                    MAIN 104
      GO TO 500                                                      MAIN 105
   80 CALL GOTO                                                      MAIN 106
      GO TO 500                                                      MAIN 107
   90 CALL ASGOTO                                                    MAIN 108
      GO TO 500                                                      MAIN 109
  100 CALL CTGOTO                                                    MAIN 110
      GO TO 500                                                      MAIN 111
```

```
130.          110 CALL ARIF                                                  MAIN 112
131.              GO TO 500                                                  MAIN 113
132.          120 CALL LOGIF                                                 MAIN 114
133.              GO TO 500                                                  MAIN 115
134.          130 CALL DO                                                    MAIN 116
135.              GO TO 500                                                  MAIN 117
136.          140 CALL CALL                                                  MAIN 118
137.              WRITE(6,66) (A(I),I=1,N)                                   MAIN 119
138.           66 FORMAT(6X,72A1)                                            MAIN 120
139.              GO TO 500                                                  MAIN 121
140.          150 IF(JJ .NE. 1) CALL ERROR(2,KDM1,KDM2)
141.              IBLKDT=1                                                   MAIN 123
142.              CALL SIMP                                                  MAIN 124
143.              GO TO 500                                                  MAIN 125
144.        C*****IF UNIVAC 1108 - PLACE A :C: IN COLUMN 1 OF NEXT 3 CARDS
145.        C 160 IF(JJ .NE. 1) CALL ERROR(2)
146.        C     CALL PROG
147.        C     GO TO 500
148.          170 DO 175 I=1,12                                              MAIN 129
149.              IF(IPREC .EQ. IORD(I)) GO TO 177                           MAIN 130
150.          175 CONTINUE                                                   MAIN 131
151.              CALL ERROR(2,KDM1,KDM2)
152.          177 CALL DIMEN                                                 MAIN 133
153.              GO TO 500                                                  MAIN 134
154.          180 DO 185 I=1,5                                               MAIN 135
155.              IF(IPREC .EQ. IORD(I)) GO TO 187                           MAIN 136
156.          185 CONTINUE                                                   MAIN 137
157.              CALL ERROR(2,KDM1,KDM2)
158.          187 CALL COM                                                   MAIN 139
159.              GO TO 500                                                  MAIN 140
160.          190 DO 195 I=1,13                                              MAIN 141
161.              IF(IPREC .EQ. IORD(I)) GO TO 197                           MAIN 142
162.          195 CONTINUE                                                   MAIN 143
163.              CALL ERROR(2,KDM1,KDM2)
164.          197 CALL EQUIV                                                 MAIN 145
165.              GO TO 500                                                  MAIN 146
166.          200 DO 205 I=1,14
167.              IF(IPREC .EQ. IORD(I)) GO TO 207
168.          205 CONTINUE
169.              CALL ERROR(2,KDM1,KDM2)
170.          207 CALL DATA
171.              GO TO 500                                                  MAIN 148
172.          210 DO 215 I=1,6                                               MAIN 149
173.              IF(IPREC .EQ. IORD(I)) GO TO 217                           MAIN 150
174.          215 CONTINUE                                                   MAIN 151
175.              CALL ERROR(2,KDM1,KDM2)
176.          217 IF(N .GT. 72) GO TO 240                                    MAIN 153
177.              WRITE(IOP,218) (A(I),I=1,N)                                MAIN 154
178.          218 FORMAT(72A1)                                               MAIN 155
179.              GO TO 250                                                  MAIN 156
180.          240 WRITE(IOP,245) (A(I),I=1,N)                                MAIN 157
181.          245 FORMAT(72A1/(5X,1H*,66A1))                                 MAIN 158
182.          250 CALL FRMAT                                                 MAIN 159
183.              GO TO 700                                                  MAIN 160
184.          220 CALL ERROR(1,KDM1,KDM2)
185.          500 CONTINUE                                                   MAIN 162
186.              IF(N .GT. 72) GO TO 540                                    MAIN 163
187.              WRITE(IOP,520) (A(I),I=1,N)                                MAIN 164
```

```
188•       520 FORMAT(72A1)                                                    MAIN 165
189•           GO TO 600                                                       MAIN 166
190•       540 WRITE(IOP,545) (A(I),I=1,N)                                     MAIN 167
191•       545 FORMAT(72A1/(5X,1H•,66A1))                                      MAIN 168
192•       600 IF(MODE .NE. 1) GO TO 700                                       MAIN 169
193•           IF(ITYP .LT. 30 .OR. ITYP .GT. 32) GO TO 700                    MAIN 170
194•           WRITE(IOP,610)                                                  MAIN 171
195•       610 FORMAT(5X,15H COMPLEX Q1COMP)                                   MAIN 172
196•           WRITE(IOP,620)                                                  MAIN 173
197•       620 FORMAT(5X,24H DOUBLE PRECISION Q1OPRE)                          MAIN 174
198•           GO TO 700
199•      2000 WRITE(IOP,2020)                                                 MAIN 176
200•      2020 FORMAT(6X,3HEND)                                                MAIN 177
201•           IF(N .NE. 72) WRITE(6,2100)                                     MAIN 178
202•      2100 FORMAT(6X,22H ILLEGAL END STATEMENT)                            MAIN 179
203•           CALL SUBCHK
204•           IF(INTR .NE. 1) GO TO 3205
205•           CALL INTRIN
206•      3205 CALL SYMTAB
207•           CALL GRT                                                        MAIN 181
208•           CALL COMCHK
209•           IF(IOVFLW .EQ. 1) GO TO 3210
210•           CALL LOOPCK                                                     MAIN 182
211•           IF(IFL .EQ. 0 .OR. IBLKDT .EQ. 1) GO TO 3210
212•           CALL FLOWCK                                                     MAIN 184
213•      3210 IF(IERR .NE. 2) GO TO 4                                         MAIN 185
214•           CALL GLOTAB                                                     MAIN 186
215•           IF(MODE .EQ. 1) GO TO 6000
216•           CALL GENROL                                                     MAIN 188
217•           ENDFILE 9
218•           REWIND 9
219•      6000 ENDFILE 8
220•           REWIND 8
221•           STOP                                                            MAIN 200
222•      7000 WRITE(6,7005)
223•      7005 FORMAT(/////5X,54H OVERFLOW OF BASIC BLOCK TABLE - PROCESSING TERM
224•          •INATED)
225•           STOP
226•           END                                                             MAIN 201
```

AUDIT Subprograms

```
      SUBROUTINE ARIF                                                    ARIF   2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),   RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                   RICH   4
      COMMON/TYP/NQQ,RHSTYP,NQ2,NQ3,LHSTYP
      COMMON/STRING/NTYPE,NSTR,STR(500)                                  ARIF   5
      COMMON/LABELS/STATRA(2,200),NLABEL                                 ARIF   6
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      INTEGER A,STATRA,STR,COMMA,BLANK,RHSTYP,AY,EF                      ARIF   8
      INTEGER BITPUT                                                     ARIF   9
      DATA LPAR/1H(/,COMMA/1H,/,BLANK/1H /,AY/1HI/,EF/1HF/               ARIF  10
      IF(NEXT(JPTR) .NE. AY) GO TO 20                                    ARIF  11
      IF(NEXT(JPTR) .NE. EF) GO TO 20                                    ARIF  12
      IF(NEXT(JPTR) .NE. LPAR) GO TO 20                                  ARIF  13
      JPTR=JPTR-1                                                        ARIF  14
      CALL EXPR                                                          ARIF  15
      NSTR=NSTR+1                                                        ARIF  16
      STR(NSTR)= -5                                                      ARIF  17
      NTYPE=1                                                            ARIF  18
      CALL PARSE                                                         ARIF  19
      CALL FNCSTR                                                        ARIF  20
      IF(RHSTYP .EQ. 1) CALL ERROR(42,KDM1,KDM2)
      CALL BLKSTR
      NBRNCH=0                                                           ARIF  22
      DO 10 I=1,3                                                        ARIF  23
      CALL GNLE                                                          ARIF  24
      IF(JTYP .NE. 5) GO TO 20                                           ARIF  25
      CALL STSRCH                                                        ARIF  26
      STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,12)                           ARIF  27
      IF( NBRNCH .EQ. 0) GO TO 5                                         ARIF  28
      DO 3 J=1,NBRNCH                                                    ARIF  29
      IF(LOC .EQ. IBLOCK(NBLOCK-J+1)) GO TO 7                            ARIF  30
    3 CONTINUE                                                           ARIF  31
    5 NBLOCK=NBLOCK+1                                                    ARIF  32
      IBLOCK(NBLOCK)=LOC                                                 ARIF  33
      NBRNCH=NBRNCH+1                                                    ARIF  34
    7 IF(I .EQ. 3) GO TO 10                                              ARIF  35
      IF(NEXT(JPTR) .NE. COMMA) GO TO 20                                 ARIF  36
   10 CONTINUE                                                           ARIF  37
      IF(NEXT(JPTR) .NE. BLANK) GO TO 20                                 ARIF  38
      NB=1                                                               ARIF  39
      RETURN                                                             ARIF  40
   20 CALL ERROR(7,KDM1,KDM2)
      RETURN                                                             ARIF  42
      END                                                                ARIF  43
```

```
      SUBROUTINE ASGOTO                                                 ASGOTO 2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      COMMON/LABELS/STATRA(2,200),NLABEL                                ASGOTO 4
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      DIMENSION IALPH(4)                                                ASGOTO 6
      INTEGER STATRA,BLANK,COMMA,RPAR,A                                 ASGOTO 7
      INTEGER BITPUT,BITGET                                             ASGOTO 8
      DATA (IALPH(I),I=1,4)/1HG,1HO,1HT,1HO/                            ASGOTO 9
      DATA BLANK/1H /,COMMA/1H,/,LPAR/1H(/,RPAR/1H)/                    ASGOTO10
      DO 5 I=1,4                                                        ASGOTO11
      IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 30                             ASGOTO12
    5 CONTINUE                                                          ASGOTO13
      CALL GNLE                                                         ASGOTO14
      IF(JTYP .NE. 2) GO TO 30                                          ASGOTO15
      CALL SEARCH                                                       ASGOTO16
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID,KDM2)
      IF(ISRCH(1) .EQ. 1) GO TO 10                                      ASGOTO18
      IDTYP=1                                                           ASGOTO19
      CALL STORE                                                        ASGOTO20
      LOC=NID                                                           ASGOTO21
   10 CALL IMPTYP                                                       ASGOTO22
      IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4) CALL ERROR(39,NXTID,KDM2)
      IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 1) CALL ERROR(19,NXTID,KDM2)
      IF(NEXT(JPTR) .NE. COMMA) GO TO 30                                ASGOTO25
      IF(NEXT(JPTR) .NE. LPAR) GO TO 30                                 ASGOTO26
      NBLOCK=NBLOCK+1                                                   ASGOTO27
      IBLOCK(NBLOCK)=5000+LOC                                           ASGOTO28
      NBRNCH=0                                                          ASGOTO29
   20 CALL GNLE                                                         ASGOTO30
      IF(JTYP .NE. 5) GO TO 30                                          ASGOTO31
      CALL STSRCH                                                       ASGOTO32
      STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,12)                          ASGOTO33
      IF(NBRNCH .EQ. 0) GO TO 25                                        ASGOTO34
      DO 22 I=1,NBRNCH                                                  ASGOTO35
      IF(LOC .EQ. IBLOCK(NBLOCK-I+1)) GO TO 27                          ASGOTO36
   22 CONTINUE                                                          ASGOTO37
   25 NBLOCK=NBLOCK+1                                                   ASGOTO38
      IBLOCK(NBLOCK)=LOC                                                ASGOTO39
      NBRNCH=NBRNCH+1                                                   ASGOTO40
   27 IF(NEXT(JPTR) .EQ. COMMA) GO TO 20                                ASGOTO41
      IF(A(JPTR-1) .NE. RPAR) GO TO 30                                  ASGOTO42
      IF(NEXT(JPTR) .NE. BLANK) GO TO 30                                ASGOTO43
      NB=1                                                              ASGOTO44
      RETURN                                                            ASGOTO45
   30 CALL ERROR(7,KDM1,KDM2)
      RETURN                                                            ASGOTO47
      END                                                               ASGOTO48
```

```
      SUBROUTINE ASSIGN                                                 ASSIGN 2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      COMMON/LABELS/STATRA(2,200),NLABEL                                ASSIGN 4
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      DIMENSION IALPH(6)                                                ASSIGN 6
      INTEGER BLANK,TEE,OH,STATRA                                       ASSIGN 7
      INTEGER BITPUT,BITGET                                             ASSIGN 8
      DATA BLANK/1H /,TEE/1HT/,OH/1HO/                                  ASSIGN 9
      DATA (IALPH(I),I=1,6)/1HA,1HS,1HS,1HI,1HG,1HN/                    ASSIGN10
      DO 5 I=1,6                                                        ASSIGN11
      IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 20                             ASSIGN12
    5 CONTINUE                                                          ASSIGN13
      CALL GNLE                                                         ASSIGN14
      IF(JTYP .NE. 5) GO TO 20                                          ASSIGN15
      CALL STSRCH                                                       ASSIGN16
      STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,12)                          ASSIGN17
      IF(NEXT(JPTR) .NE. TEE) GOTO 20                                   ASSIGN18
      IF(NEXT(JPTR) .NE. OH) GO TO 20                                   ASSIGN19
      CALL GNLE                                                         ASSIGN20
      IF(JTYP .NE. 2) GO TO 20                                          ASSIGN21
      CALL SEARCH                                                       ASSIGN22
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID,KDM2)
      IF(ISRCH(1) .EQ. 1) GO TO 10                                      ASSIGN24
      IDTYP=1                                                           ASSIGN25
      CALL STORE                                                        ASSIGN26
      LOC=NID                                                           ASSIGN27
   10 CALL IMPTYP                                                       ASSIGN28
      IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4) CALL ERROR(39,NXTID,KDM2)
      IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 1) CALL ERROR(14,NXTID,KDM2)
      IF(NEXT(JPTR) .NE. BLANK) GO TO 20                                ASSIGN31
      NBLOCK=NBLOCK+1                                                   ASSIGN32
      IBLOCK(NBLOCK)=4000+LOC                                           ASSIGN33
      RETURN                                                            ASSIGN34
   20 CALL ERROR(7,KDM1,KDM2)
      RETURN                                                            ASSIGN36
      END                                                               ASSIGN37
```

```
      SUBROUTINE AUXIO                                                  AUXIO  2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      DIMENSION IALPH1(6),IALPH2(9),IALPH3(7)                           AUXIO  5
      INTEGER BITGET                                                    AUXIO  6
      DATA (IALPH1(I),I=1,6)/1HR,1HE,1H#,1HI,1HN,1HD/                   AUXIO  7
      DATA (IALPH2(I),I=1,9)/1HB,1HA,1HC,1HK,1HS,1HP,1HA,1HC,1HE/       AUXIO  8
      DATA (IALPH3(I),I=1,7)/1HE,1HN,1HD,1HF,1HI,1HL,1HE/               AUXIO  9
      IT=16-ITYP                                                        AUXIO 10
      GO TO (25,15,5),IT                                                AUXIO 11
    5 DO 10 I=1,6                                                       AUXIO 12
      IF(NEXT(JPTR) .NE. IALPH1(I)) GO TO 50                            AUXIO 13
   10 CONTINUE                                                          AUXIO 14
      GO TO 40                                                          AUXIO 15
   15 DO 20 I=1,9                                                       AUXIO 16
      IF(NEXT(JPTR) .NE. IALPH2(I)) GO TO 50                            AUXIO 17
   20 CONTINUE                                                          AUXIO 18
      GO TO 40                                                          AUXIO 19
   25 DO 30 I=1,7                                                       AUXIO 20
      IF(NEXT(JPTR) .NE. IALPH3(I)) GO TO 50                            AUXIO 21
   30 CONTINUE                                                          AUXIO 22
   40 CALL GNLE                                                         AUXIO 23
      IF(JTYP .NE. 2) GO TO 60                                          AUXIO 24
      IF(NEXT(JPTR) .NE. 1H ) GO TO 50                                  AUXIO 25
      CALL SEARCH                                                       AUXIO 26
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID,KDM2)
      IF(ISRCH(1) .EQ. 1) GO TO 45                                      AUXIO 28
      IDTYP=1                                                           AUXIO 29
      CALL STORE                                                        AUXIO 30
      LOC=NID                                                           AUXIO 31
   45 CALL IMPTYP                                                       AUXIO 32
      IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4) CALL ERROR(22,KDM1,KDM2)
      IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 1) CALL ERROR(14,NXTID,KDM2)
      NBLOCK=NBLOCK+1                                                   AUXIO 35
      IBLOCK(NBLOCK)=2000+LOC                                           AUXIO 36
      RETURN                                                            AUXIO 37
   50 CALL ERROR(7,KDM1,KDM2)
      RETURN                                                            AUXIO 39
   60 CALL ERROR(22,KDM1,KDM2)
      RETURN                                                            AUXIO 41
      END                                                               AUXIO 42
```

```
      COMPILER(FLD=ABS)
      INTEGER FUNCTION BITGET(ILOC,IPOS,I#IDTH)
      ISTART=IPOS-I#IDTH
      BITGET=FLD(ISTART,I#IDTH,ILOC)
      RETURN
      END
```

```
      COMPILER(FLD=ABS)
      INTEGER FUNCTION BITPUT(ILOC,IVAL,IPOS)
      JLOC=ILOC
      DO 10 I#IDTH=1,36
      IF(IVAL .GE. 2**I#IDTH) GO TO 10
      IBIT=IPOS-I#IDTH
      GO TO 15
   10 CONTINUE
   15 FLD(IBIT,I#IDTH,JLOC)=IVAL
      BITPUT=JLOC
      RETURN
      END
```

```
      SUBROUTINE BLKSTR
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/FUNC/IFNCRA(5,17),MARGS,IARGS(50),FNCLOC(5),NFUNC
      COMMON/LIST/NLIST,NINTFC,ISUBLT(3,200),INTFAC(500)
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      INTEGER BITPUT,BITGET,FNCLOC
      IF(MARGS .EQ. 0) RETURN
      DO 100 I=1,MARGS
      IOSTAT=2
      ICOL=18*MOD(I-1,2)+9
      IVR=(I+1)/2
      LOC=BITGET(IARGS(IVR),ICOL,9)
      NFNC=BITGET(IARGS(IVR),ICOL+3,3)
      IF(NFNC .EQ. 0) GO TO 60
      ILOC=FNCLOC(NFNC)
      NARG=BITGET(IARGS(IVR),ICOL+9,6)
      INDEX=BITGET(IDTBL(3,ILOC),36,9)
      IF(INDEX .EQ. 0) GO TO 60
      IPTR=ISUBLT(3,INDEX)+(NARG-1)/4
      JVAR=BITGET(ISUBLT(2,INDEX ),14,1)
      ICOL=9*MOD(NARG-1,4)+8
      IF(JVAR .EQ. 1) ICOL=8
      IOSTAT=BITGET(INTFAC(IPTR),ICOL,2)
      KPTR=(NARG+7)/4
      IEXP=BITGET(IFNCRA(NFNC,KPTR),ICOL+1,1)
      IF(IOSTAT .EQ. 2) GO TO 60
      IF(IEXP .NE. 0) GO TO 40
      IF(IOSTAT .EQ. 1) GO TO 60
      GO TO 80
   40 IF(BITGET(ISUBLT(2,INDEX),10,4) .NE. 0) GO TO 90
      INTFAC(IPTR)=BITPUT(INTFAC(IPTR),2,ICOL)
      IOSTAT=2
   60 NBLOCK=NBLOCK+1
      IBLOCK(NBLOCK)=2000+LOC
      IF(IOSTAT .EQ. 2) GO TO 100
   80 NBLOCK=NBLOCK+1
      IBLOCK(NBLOCK)=1000+LOC
      GO TO 100
   90 CALL ERROR(55,NARG,KDM2)
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE BUILD                                                  BUILD  2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      COMMON/INPOUT/NCALL,IN,IOP                                        BUILD  4
      INTEGER A,B,BLANK                                                 BUILD  5
      COMMON/WASTE/B(72)
      DATA BLANK/1H /,ICE/1HC/                                          BUILD  7
      IERR=0                                                            BUILD  8
      NFIRST=1                                                          BUILD  9
      NCONTU=0                                                          BUILD 10
      NCALL=NCALL+1                                                     BUILD 11
   50 CONTINUE
      IF(NFIRST .EQ. 1 .AND. NCALL .NE. 1) GO TO 1                      BUILD 13
C****IF UNIVAC 1108 - PLACE A 'C' IN COLUMN 1 OF NEXT TWO CARDS         CH34  21
C     READ(IN,100) (B(I),I=1,72)
C     IF(EOF(IN) .NE. 0) GO TO 10
C****IF CDC 6700 - PLACE A 'C' IN COLUMN 1 OF NEXT CARD                 CH34  23
      READ(IN,100,END=10) (B(I),I=1,72)
  100 FORMAT(72A1)                                                      BUILD 15
    1 CONTINUE                                                          BUILD 17
      IF(NFIRST .EQ. 1) GO TO 2                                         BUILD 18
      IF(B(1) .EQ. ICE) GO TO 9                                         BUILD 19
      IF(B(6) .NE. BLANK .AND. B(6) .NE. 0) GO TO 6                     BUILD 20
      GO TO 9                                                           BUILD 21
    2 CONTINUE                                                          BUILD 22
      DO 3 I=1,72                                                       BUILD 23
      A(I)=B(I)                                                         BUILD 24
    3 CONTINUE                                                          BUILD 25
      NFIRST=0                                                          BUILD 26
      NCHAR=72
      GO TO 50
    6 NCONTU=NCONTU+1                                                   BUILD 28
      IF(NCONTU .LE. 19) GO TO 7                                        BUILD 29
      IERR=1                                                            BUILD 30
      CALL ERROR(4,KDM1,KDM2)
      RETURN                                                            BUILD 32
    7 CONTINUE                                                          BUILD 33
      DO 8 I=1,66                                                       BUILD 34
    8 A(NCHAR+I)=B(I+6)
      NCHAR=NCHAR+66
      GO TO 50
   10 IERR=2                                                            BUILD 39
    9 CONTINUE                                                          BUILD 40
      N=NCHAR
      RETURN                                                            BUILD 42
      END                                                               BUILD 44
```

```
1•          COMPILER(FLD=ABS)
2•          SUBROUTINE CAA(ISTR,MSTR,ID)
3•          DIMENSION ISTR(6)
4•          IF(MSTR .GT. 6 .AND. ITYP .NE. 28) CALL ERROR(6,KDM1,KDM2)
5•          ID=1H
6•          DO 20 I=1,MSTR
7•          IBIT=6*(I-1)
8•          ICHAR=FLD(0,6,ISTR(I))
9•       20 FLD(IBIT,6,ID)=ICHAR
10•         RETURN
11•         END
```

```
1•          SUBROUTINE CAI(ISTR,MSTR,INTVAL)
2•          DIMENSION ISTR(10)
3•          INTEGER BITGET
4•          IF(MSTR .GT. 10) CALL ERROR(3,KDM1,KDM2)
5•          INTVAL=0
6•          DO 10 I=1,MSTR
7•          INT=BITGET(ISTR(I),6,6)-48
8•          IF(INT .EQ. 0) GO TO 10
9•          INTVAL=INTVAL+INT*10**(MSTR-I)
10•      10 CONTINUE
11•         IF(INTVAL .GT. (2**31-1)) CALL ERROR(3,KDM1,KDM2)
12•         RETURN
13•         END
```

```
      SUBROUTINE CALL                                                    CALL   2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),   RICH   2
     • JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                   RICH   4
      COMMON/STRING/NTYPE,NSTR,STR(500)                                  CALL   4
      COMMON/LIST/NLIST,NINTFC,ISUBLT(3,200),INTFAC(500)
      DIMENSION IALPH(4)
      INTEGER BLANK,BITPUT,BITGET
      DATA (IALPH(I),I=1,4)/1HC,1HA,1HL,1HL/                             CALL  11
      DATA LPAR/1H(/,BLANK/1H /
      DO 5 I=1,4                                                         CALL  14
      IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 50                              CALL  15
    5 CONTINUE                                                           CALL  16
      IPTR=JPTR
    7 CALL GNLE                                                          CALL  17
      IF(JTYP .NE. 2) GO TO 50                                           CALL  18
      IF(NXTID .EQ. IFNCNM) CALL ERROR(10,NXTID,KDM1)
      CALL SEARCH                                                        CALL  20
      IF(ISRCH(1) .EQ. 1) CALL ERROR(24,NXTID,KDM2)
      IF(ISRCH(2) .EQ. 1) GO TO 8                                        CALL  22
      IDTYP=2                                                            CALL  23
      CALL STORE                                                         CALL  24
      LOC=NID                                                            CALL  25
    8 CONTINUE                                                           CALL  26
      ILOC=LOC
      NXT=NEXT(JPTR)                                                     CALL  32
      IF(NXT .EQ. LPAR) GO TO 17
      IF(NXT .NE. BLANK) GO TO 50                                        CALL  34
      IF(BITGET(IDTBL(3,LOC),18,1) .EQ. 1) GO TO 20
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,18)
      DO 10 I=1,NLIST
      IF(IDTBL(1,LOC) .NE. ISUBLT(1,I)) GO TO 10
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,36)
      LISTLC=I
      GO TO 22
   10 CONTINUE
      NLIST=NLIST+1
      ISUBLT(1,NLIST)=IDTBL(1,LOC)
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),NLIST,36)
      CALL ERROR(52,KDM1,KDM2)
      RETURN
   20 LISTLC=BITGET(IDTBL(3,LOC),36,9)
   22 CONTINUE
      IF(BITGET(ISUBLT(2,LISTLC),6,6) .NE. 0) CALL ERROR(26,KDM1,KDM2)
      RETURN
   17 JPTR=IPTR
      NTYPE=1
      CALL EXPR                                                          CALL  69
      CALL PARSE                                                         CALL  70
      CALL FNCSTR                                                        CALL  71
      CALL BLKSTR
   47 IF(MODE .EQ. 1) GO TO 48
      LOC=ILOC
      INDEX=BITGET(IDTBL(3,LOC),36,9)
      KLAS=BITGET(ISUBLT(2,INDEX),10,4)
      IF(KLAS .EQ. 1 .OR. KLAS .EQ. 2) CALL CALL2
      RETURN
   48 JPTR=IPTR-1
      CALL CNVRT
      RETURN                                                             CALL  98
   50 CALL ERROR(7,KDM1,KDM2)
      RETURN                                                             CALL 100
      END                                                                CALL 101
```

```
      SUBROUTINE CALL2                                                  CALL2  2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      INTEGER A,D,COMMA,RPAR                                            CALL2  5
      INTEGER BITPUT,BITGET                                             CALL2  6
      DIMENSION IALPH(13)                                               CALL2  7
      DATA RPAR/1H)/,COMMA/1H,/                                         CALL2  8
      DATA (IALPH(I),I=1,13)/1HC,1HA,1HL,1HL,1H ,1HR,1HO,1HL,1HC,1HM,   CALL2  9
     1 1HK,1H ,1H(/                                                     CALL2 10
      DO 15 J=1,13                                                      CALL2 11
      K=J+6                                                             CALL2 12
      A(K)=IALPH(J)                                                     CALL2 13
   15 CONTINUE                                                          CALL2 14
      DO 20 I=1,6                                                       CALL2 15
      KK=19+4*I                                                         CALL2 16
      A(KK-3)=1H1                                                       CALL2 17
      A(KK-2)=1HH                                                       CALL2 18
      IPOS=6*I                                                          CALL2 19
      IVL=BITGET(IDTBL(1,LOC),IPOS,6)
      A(KK-1)=BITPUT(0,IVL,6)                                           CH33   2
      IF(I .EQ. 6) GO TO 25                                             CALL2 22
   20 A(KK)=COMMA                                                       CALL2 23
   25 A(KK)=RPAR                                                        CALL2 24
      N=KK                                                              CALL2 25
      RETURN                                                            CALL2 26
      END                                                               CALL2 27
```

```
      SUBROUTINE CHKLST                                                 CH34  36
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      DIMENSION IQUIV(100)                                              CHKLIST4
      EQUIVALENCE (IQUIV(1),A(301))
      INTEGER BITGET                                                    CHKLIST5
      NQUIV=0                                                           CHKLIST6
      DO 30 LOC=1,NID                                                   CHKLIST7
      IDTBL(8,LOC)=0
      IF(BITGET(IDTBL(3,LOC),14,1) .EQ. 1) GO TO 20                     CHKLIST8
      IF(BITGET(IDTBL(3,LOC),16,1) .EQ. 1) GO TO 5                      CHKLIST9
      IF(BITGET(IDTBL(3,LOC),12,1) .EQ. 1) GO TO 15
      GO TO 10                                                          CHKLIS12
    5 IDTBL(8,LOC)=1                                                    CHKLIS13
   10 IF(BITGET(IDTBL(3,LOC),[illegible],1) .NE. 1) GO TO 30            CHKLIS14
      NQUIV=NQUIV+1                                                     CHKLIS15
      IF(NQUIV .GT. 100) GO TO 60
      IQUIV(NQUIV)=LOC                                                  CHKLIS16
      GO TO 30                                                          CHKLIS17
   15 IF(BITGET(IDTBL(3,LOC),15,1) .EQ. 0) GO TO 30
   20 IDTBL(8,LOC)=1                                                    CHKLIS18
   30 CONTINUE                                                          CHKLIS19
      IF(NQUIV .EQ. 0) RETURN                                           CHKLIS20
      DO 50 J=1,NQUIV                                                   CHKLIS21
      NXQV=IQUIV(J)                                                     CHKLIS22
   35 NXQV=IDTBL(7,NXQV)                                                CHKLIS23
      IF(NXQV .EQ. IQUIV(J)) GO TO 50                                   CHKLIS24
      IF(IDTBL(8,NXQV) .EQ. 0) GO TO 35                                 CHKLIS25
      IQV=NXQV                                                          CHKLIS26
      KTYPE=BITGET(IDTBL(3,IQV),10,3)                                   CHKLIS27
   40 NXQV=IDTBL(7,NXQV)                                                CHKLIS28
      IF(NXQV .EQ. IQV) GO TO 50                                        CHKLIS29
      IF(BITGET(IDTBL(3,NXQV),10,3) .NE. KTYPE) GO TO 40                CHKLIS30
      IDTBL(8,NXQV)=1                                                   CHKLIS31
      GO TO 40                                                          CHKLIS32
   50 CONTINUE                                                          CHKLIS33
      RETURN                                                            CHKLIS34
   60 CALL ERROR(94,IDM1,IDM2)
      RETURN
      END                                                               CHKLIS35
```

```
      SUBROUTINE CLASS                                                  CLASS  2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      DIMENSION KALP(48),KSUC(48),KFAL(48),KDEC(10),KF(8)               CLASS  4
      INTEGER A                                                         CLASS  5
      DATA KDEC(1),KDEC(2),KDEC(3),KDEC(4),KDEC(5),                    CLASS  6
     1  KDEC(6),KDEC(7),KDEC(8),KDEC(9),KDEC(10)                       CLASS  7
     2  /1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/                      CLASS  8
      DATA KF(1),KF(2),KF(3),KF(4),KF(5),KF(6),KF(7),KF(8)             CLASS  9
     1   /1HF,1HU,1HN,1HC,1HT,1HI,1HO,1HN/                            CLASS 10
      DATA KC,KBLNK,KLPAR,KRPAR,KEQ  /1HC,1H ,1H(,1H),1H=/             CLASS 11
      DATA KH,KSLSH,KASTK,KZERO,KCMA  /1HH,1H/,1H*,1H0,1H,/            CLASS 12
      DATA KALP( 1),KALP( 2),KALP( 3),KALP( 4) /1HI,1HF,1HN,1HG/       CLASS 13
      DATA KALP( 5),KALP( 6),KALP( 7),KALP( 8) /1H0,1HT,1H0,1H(/       CLASS 14
      DATA KALP( 9),KALP(10),KALP(11),KALP(12) /1HC,1HA,1H0,1HN/       CLASS 15
      DATA KALP(13),KALP(14),KALP(15),KALP(16) /1HM,1HM,1HP,1HR/       CLASS 16
      DATA KALP(17),KALP(18),KALP(19),KALP(20) /1HE,1HA,1HD,1HL/       CLASS 17
      DATA KALP(21),KALP(22),KALP(23),KALP(24) /1HT,1H*,1HF,1H0/       CLASS 18
      DATA KALP(25),KALP(26),KALP(27),KALP(28) /1HU,1HD,1HI,1HA/       CLASS 19
      DATA KALP(29),KALP(30),KALP(31),KALP(32) /1H0,1HU,1H*,1HS/       CLASS 20
      DATA KALP(33),KALP(34),KALP(35),KALP(36) /1HT,1HU,1HE,1HN/       CLASS 21
      DATA KALP(37),KALP(38),KALP(39),KALP(40) /1HD,1HF,1HX,1HQ/       CLASS 22
      DATA KALP(41),KALP(42),KALP(43),KALP(44) /1HB,1HA,1HL,1HA/       CLASS 23
      DATA KALP(45),KALP(46),KALP(47),KALP(48)/1HL,1HP,1HR,1H0/        CLASS 24
      DATA KSUC( 1),KSUC( 2),KSUC( 3),KSUC( 4) /  2, -6,-19,  5/       CLASS 25
      DATA KSUC( 5),KSUC( 6),KSUC( 7),KSUC( 8) /  6,  7,  8, -5/       CLASS 26
      DATA KSUC( 9),KSUC(10),KSUC(11),KSUC(12) / 10, -8, 12, -7/       CLASS 27
      DATA KSUC(13),KSUC(14),KSUC(15),KSUC(16) / 14,-25,-22, 17/       CLASS 28
      DATA KSUC(17),KSUC(18),KSUC(19),KSUC(20) / 18, 19,-11,-20/       CLASS 29
      DATA KSUC(21),KSUC(22),KSUC(23),KSUC(24) / -9,-13, 24,-28/       CLASS 30
      DATA KSUC(25),KSUC(26),KSUC(27),KSUC(28) /-31, 27,-24,-27/       CLASS 31
      DATA KSUC(29),KSUC(30),KSUC(31),KSUC(32) / 30,-21,-12, 33/       CLASS 32
      DATA KSUC(33),KSUC(34),KSUC(35),KSUC(36) /-10,-30, 36, 37/       CLASS 33
      DATA KSUC(37),KSUC(38),KSUC(39),KSUC(40) / 38,-15,-34,-26/       CLASS 34
      DATA KSUC(41),KSUC(42),KSUC(43),KSUC(44) / 42,-14,-29, -2/       CLASS 35
      DATA KSUC(45),KSUC(46),KSUC(47),KSUC(48) /-23,47,48,-32/
      DATA KFAL( 1),KFAL( 2),KFAL( 3),KFAL( 4) /  4,  3,-36,  9/       CLASS 37
      DATA KFAL( 5),KFAL( 6),KFAL( 7),KFAL( 8) /-36,-36,-36, -3/       CLASS 38
      DATA KFAL( 9),KFAL(10),KFAL(11),KFAL(12) / 16, 11,-36, 13/       CLASS 39
      DATA KFAL(13),KFAL(14),KFAL(15),KFAL(16) /-36, 15,-36, 23/       CLASS 40
      DATA KFAL(17),KFAL(18),KFAL(19),KFAL(20) /-36, 21, 20,-36/       CLASS 41
      DATA KFAL(21),KFAL(22),KFAL(23),KFAL(24) / 22,-36, 26, 25/       CLASS 42
      DATA KFAL(25),KFAL(26),KFAL(27),KFAL(28) /-36, 31, 28, 29/       CLASS 43
      DATA KFAL(29),KFAL(30),KFAL(31),KFAL(32) /-36,-36, 32, 35/       CLASS 44
      DATA KFAL(33),KFAL(34),KFAL(35),KFAL(36) / 34,-36, 41, 39/       CLASS 45
      DATA KFAL(37),KFAL(38),KFAL(39),KFAL(40) /-36,-18, 40,-36/       CLASS 46
      DATA KFAL(41),KFAL(42),KFAL(43),KFAL(44) / 44, 43,-36, 45/       CLASS 47
      DATA KFAL(45),KFAL(46),KFAL(47),KFAL(48)/ 46,-36,-36,-36/        CLASS 48
      LTYP=0                                                           CLASS 49
      IPTR=7                                                           CLASS 50
    5 CONTINUE                                                         CLASS 51
      JSAVE=KBLNK                                                      CLASS 52
      JS#=0                                                            CLASS 53
      IS#=0                                                            CLASS 54
      JEQ=0                                                            CLASS 55
      JCMA=0                                                           CLASS 56
      JHOLL=0                                                          CLASS 57
      DO 26 J=IPTR,N                                                   CLASS 58
      JCH=A(J)                                                         CLASS 59
      IF(JCH .EQ. KBLNK) GO TO 26                                      CLASS 60
    6 IF(JHOLL .LE. 0) GO TO 12                                        CLASS 61
    7 DO 8 L=1,10                                                      CLASS 62
      IF(JCH .EQ. KDEC(L)) GO TO 10                                    CLASS 63
    8 CONTINUE                                                         CLASS 64
      IF(JHOLL .LE. 11) GO TO 11                                       CLASS 65
    9 IF(JCH-KH) 11,32,11                                              CLASS 66
   10 JHOLL=JHOLL+1                                                    CLASS 67
      GO TO 25                                                         CLASS 68
   11 JHOLL=0                                                          CLASS 69
```

```
      12 IF(JCH .EQ. KLPAR) GO TO 20                                    CLASS 70
      13 IF(JCH .EQ. KRPAR) GO TO 18                                    CLASS 71
      14 IF(JCH .EQ. KCMA) GO TO 22                                     CLASS 72
      15 IF(JCH .EQ. KEQ) GO TO 23                                      CLASS 73
      16 IF(JCH .EQ. KSLSH) GO TO 21                                    CLASS 74
      17 IF(JCH .EQ. KASTK) 25,21,25                                    CLASS 75
      18 JSW=JSW-1                                                      CLASS 76
         IF(JSW .GT. 0) GO TO 25                                        CLASS 77
      19 ISW=1                                                          CLASS 78
         GO TO 26                                                       CLASS 79
      20 JSW=JSW+1                                                      CLASS 80
      21 JHOLL=1                                                        CLASS 81
         GO TO 25                                                       CLASS 82
      22 IF(JSW) 30,30,21                                               CLASS 83
      23 IF(JSW .GT. 0) GO TO 32                                        CLASS 84
      24 JEQ=1                                                          CLASS 85
      25 IF(ISW .GT. 0) GO TO 27                                        CLASS 86
      26 CONTINUE                                                       CLASS 87
         GO TO 28                                                       CLASS 88
      27 JSAVE=JCH                                                      CLASS 89
         JP=J                                                           CLASS 90
      28 IF(JEQ .LE. 0) GO TO 32                                        CLASS 91
      29 JTP=1                                                          CLASS 92
         GO TO 55                                                       CLASS 93
      30 JCMA=1                                                         CLASS 94
         IF(JEQ .LE. 0) GO TO 32                                        CLASS 95
      31 JTP=17                                                         CLASS 96
         GO TO 55                                                       CLASS 97
      32 J=1                                                            CLASS 98
         ISW=IPTR                                                       CLASS 99
      33 JCH=A(ISW)                                                     CLASS100
         IF(JCH .EQ. KBLNK) GO TO 37                                    CLASS101
      34 IF(JCH .EQ. KALP(J)) GO TO 36                                  CLASS102
      35 J=KFAL(J)                                                      CLASS103
         IF (J) 39,39,34                                                CLASS104
      36 J=KSUC(J)                                                      CLASS105
         IF(J .LE. 0) GO TO 39                                          CLASS106
      37 ISW=ISW+1                                                      CLASS107
         IF(ISW .LE. N) GO TO 33                                        CLASS108
      38 JCH=KBLNK                                                      CLASS109
         GO TO 35                                                       CLASS110
      39 JTP=-J                                                         CLASS111
         IF(JTP-3) 55,45,40                                             CLASS112
      40 IF(JTP-6) 55,43,41                                             CLASS113
      41 IF(JTP .LT. 19) GO TO 55                                       CLASS114
      42 IF(JTP - 23) 47,47,55                                          CLASS115
      43 DO 44 L=1,10                                                   CLASS116
         IF(JSAVE .EQ. KDEC(L)) GO TO 55                                CLASS117
      44 CONTINUE                                                       CLASS118
         LTYP=9                                                         CLASS119
         JTP=16                                                         CLASS120
         IPTR=JP                                                        CLASS121
         GO TO 5                                                        CLASS122
      45 IF(JCMA .LE. 0) GO TO 55                                       CLASS123
         JTP=4                                                          CLASS124
         GO TO 55                                                       CLASS125
      47 L=11                                                           CLASS126
         GO TO 52                                                       CLASS127
      48 L=L+1                                                          CLASS128
         IF(L .GT. N) GO TO 55                                          CLASS129
      49 IF(A(L) .EQ. KBLNK) GO TO 48                                   CLASS130
      50 IF(A(L) .EQ. KF(ISW)) GO TO 53                                 CLASS131
      51 IF(ISW .EQ. 1) GO TO 48                                        CLASS132
      52 ISW=1                                                          CLASS133
         GO TO 50                                                       CLASS134
      53 ISW=ISW+1                                                      CLASS135
         IF(ISW .LE. 8) GO TO 48                                        CLASS136
      54 JTP=31                                                         CLASS137
      55 ITYP=JTP                                                       CLASS138
         RETURN                                                         CLASS139
         END                                                            CLASS140
```

```
      SUBROUTINE CMPARE
      DIMENSION IROLCK(100),ISUB(100)
      REWIND 3
      NSUB=0
      NC=0
      DO 5 I=1,100
      READ(9,END=7) ISUB(I)
      NSUB=NSUB+1
    5 CONTINUE
    7 IF(NSUB .EQ. 0) RETURN
      DO 40 L=1,12
      MODE=L-1
      NROLL=0
      DO 10 I=1,100
      READ(3,END=15) IROLCK(I)
      NROLL=NROLL+1
   10 CONTINUE
   15 IF(NROLL .EQ. 0) GO TO 35
      DO 30 J=1,NSUB
      DO 20 K=1,NROLL
      IF(IROLCK(K) .EQ. ISUB(J)) GO TO 30
   20 CONTINUE
      NC=1
      WRITE(6,25) ISUB(J),MODE
   25 FORMAT(/29X,12H SUBROUTINE ,A6,53H WAS NOT CALLED IN THE ROLL CALL
     * MODE FOR MODE INDEX ,I3)
   30 CONTINUE
      GO TO 40
   35 WRITE(6,36) MODE
   36 FORMAT(/32X,65H NO SUBROUTINES WERE CALLED IN THE ROLL CALL MODE F
     *OR MODE INDEX ,I3)
      NC=1
   40 CONTINUE
      IF(NC .EQ. 0) WRITE(6,50)
   50 FORMAT(/41X,50H ALL SUBROUTINES WERE CALLED IN THE ROLL CALL MODE)
      RETURN
      END
```

```
      SUBROUTINE CNVRT                                                  CNVRT  2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      COMMON/STRING/NTYPE,NSTR,STR(500)                                 CNVRT  4
      DIMENSION IOP(8),ILOG(7,2),JLOG(2,3),IFUNC1(6),IFUNC2(6),         CNVRT  6
     1 IFUNC3(6)                                                        CNVRT  7
      INTEGER A,STR,D,DECPT
      INTEGER BITPUT,BITGET                                             CNVRT  9
      DATA (IOP(I),I=1,8)/1H+,1H-,1H/,1H(,1H),1H,,1H*,1H=/              CNVRT 10
      DATA (((ILOG(I,J),J=1,2),I=1,7)/1HL,1HT,1HL,1HE,1HG,1HT,1HG,1HE,  CNVRT 11
     1 1HE,1HQ,1HN,1HE,1HO,1HR/                                         CNVRT 12
      DATA ((JLOG(I,J),J=1,3),I=1,2)/1HA,1HN,1HD,1HN,1HO,1HT/           CNVRT 13
      DATA (IFUNC1(I),I=1,6)/1HQ,1H1,1HR,1HE,1HA,1HL/                   CNVRT 14
      DATA (IFUNC2(I),I=1,6)/1HQ,1H1,1HD,1HP,1HR,1HE/                   CNVRT 15
      DATA (IFUNC3(I),I=1,6)/1HQ,1H1,1HC,1HO,1HM,1HP/                   CNVRT 16
      DATA DECPT/1H./
      DO 5 J=1,JPTR
    5 D(J)=A(J)
      J=JPTR
      DO 100 K=1,NSTR                                                   CNVRT 27
      IF(STR(K) .GT. 0) GO TO 40
      DO 10 I=1,8                                                       CNVRT 28
      IF(STR(K) .NE. -I) GO TO 10                                       CNVRT 29
      D(J+1)=IOP(I)
      N3=1                                                              CNVRT 31
      IF(I .NE. 8) GO TO 100                                            CNVRT 32
      D(J+2)=IOP(I)
      N3=2                                                              CNVRT 34
      GO TO 100                                                         CNVRT 35
   10 CONTINUE                                                          CNVRT 36
      DO 15 I=1,7                                                       CNVRT 37
      L=I+8                                                             CNVRT 38
      IF(STR(K) .NE. -L) GO TO 15                                       CNVRT 39
      D(J+1)=DECPT
      D(J+2)=ILOG(I,1)
      D(J+3)=ILOG(I,2)
      D(J+4)=DECPT
      N3=4                                                              CNVRT 44
      GO TO 100                                                         CNVRT 45
   15 CONTINUE                                                          CNVRT 46
      DO 20 I=1,2                                                       CNVRT 47
      L=I+15                                                            CNVRT 48
      IF(STR(K) .NE. -L) GO TO 20                                       CNVRT 49
      D(J+1)=DECPT
      D(J+2)=JLOG(I,1)
      D(J+3)=JLOG(I,2)
      D(J+4)=JLOG(I,3)
      D(J+5)=DECPT
      N3=5                                                              CNVRT 55
      GO TO 100                                                         CNVRT 56
   20 CONTINUE                                                          CNVRT 57
      KL=1                                                              CNVRT 59
      IF(STR(K) .EQ. -0) KL=2                                           CNVRT 60
      IF(STR(K) .EQ. -10000) KL=4                                       CNVRT 61
      IF(STR(K) .EQ. -20000) KL=3                                       CNVRT 62
      GO TO(110,25,30,35),KL                                            CNVRT 63
   25 DO 27 I=1,6                                                       CNVRT 64
      D(J+1)=IFUNC1(I)
   27 CONTINUE                                                          CNVRT 66
      N3=6                                                              CNVRT 67
      GO TO 100                                                         CNVRT 68
   30 DO 32 I=1,6                                                       CNVRT 69
      D(J+1)=IFUNC2(I)
   32 CONTINUE                                                          CNVRT 71
      N3=6                                                              CNVRT 72
      GO TO 100                                                         CNVRT 73
   35 DO 37 I=1,6                                                       CNVRT 74
      D(J+1)=IFUNC3(I)
   37 CONTINUE                                                          CNVRT 76
      N3=6                                                              CNVRT 77
```

```
      GO TO 100                                                        CNVRT 78
   40 IF(STR(K) .LT. 1000001) GO TO 110                                CNVRT 79
      N3=STR(K)/1.E6                                                   CNVRT 80
      NLOC=(STR(K)/10000)*10000                                        CNVRT 81
      JPTR=STR(K)-NLOC                                                 CNVRT 82
      KLOC=STR(K)-N3*1.E6                                              CNVRT 83
      IF(KLOC .LT. 400000 .OR. KLOC .GT. 500000) GO TO 50              CNVRT 84
      IHL=(KLOC-400000)/10000
      IF(IHL .EQ. 5) GO TO 60
      DO 45 I=1,N3                                                     CNVRT 85
      D(J+I)=NEXT(JPTR)
   45 CONTINUE                                                         CNVRT 87
      GO TO 100                                                        CNVRT 88
   50 DO 55 I=1,N3                                                     CNVRT 89
      IPOS=6*I                                                         CNVRT 90
      ICHAR=BITGET(IDTBL(1,JPTR),IPOS,6)                               CNVRT 91
   55 D(J+I)=BITPUT(0,ICHAR,6)
      GO TO 100
   60 KPTR=JPTR-1
      DO 65 I=1,N3
   65 D(J+I)=A(KPTR+I)
  100 J=J+N3                                                           CNVRT 93
      N=J
      DO 105 I=1,N
  105 A(I)=D(I)
      RETURN                                                           CNVRT100
  110 CALL ERROR(23,KDM1,KDM2)
      RETURN                                                           CNVRT102
      END                                                              CNVRT103
```

```
      SUBROUTINE COM                                                    COM    2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     • JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      DIMENSION IDIM(3),IALPH(6)                                        COM    4
      INTEGER SLASH,COMMA,BLANK,A,RPAR                                  COM    5
      INTEGER BITPUT,BITGET                                             COM    6
      DATA (IALPH(I),I=1,6)/1HC,1HO,1HM,1HM,1HO,1HN/                    COM    7
      DATA SLASH/1H//,COMMA/1H,/,BLANK/1H /,RPAR/1H)/,LPAR/1H(/         COM    8
      DO 10 I=1,6                                                       COM    9
      IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 60                             COM   10
   10 CONTINUE                                                          COM   11
      IF(NEXT(JPTR) .EQ. SLASH) GO TO 15                                COM   12
      JPTR=JPTR-1                                                       COM   13
   12 NXTID=BLANK                                                       COM   14
      GO TO 20                                                          COM   15
   15 CALL GNLE                                                         COM   16
      IF(A(JPTR-1) .EQ. SLASH) GO TO 12                                 COM   17
      IF(JTYP .NE. 2) GO TO 60                                          COM   18
      IF(NEXT(JPTR) .NE. SLASH) GO TO 60                                COM   19
   20 CALL COMSCH                                                       COM   20
      IF(ISRCH(3) .EQ. 1) GO TO 25                                      COM   21
      IDTYP=3                                                           COM   22
      CALL STORE                                                        COM   23
      ICMLOC=NID                                                        COM   24
      GO TO 27                                                          COM   25
   25 ICMLOC=LOC                                                        COM   26
      LSTLOC=IDTBL(6,ICMLOC)                                            COM   27
   27 CALL GNLE                                                         COM   28
      IF(JTYP .NE. 2) GO TO 60                                          COM   29
      CALL SEARCH                                                       COM   30
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID,KDM2)
      IF(ISRCH(1) .EQ. 1) GO TO 28                                      COM   32
      IDTYP=1                                                           COM   33
      CALL STORE                                                        COM   34
      LOC=NID                                                           COM   35
   28 IF(BITGET(IDTBL(3,LOC),12,1) .EQ. 1) CALL ERROR(17,NXTID,KDM2)
      IF(BITGET(IDTBL(3,LOC),16,1) .EQ. 1) CALL ERROR(53,NXTID,KDM2)
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,16)                            COM   37
      ICMSIZ=1                                                          COM   38
      IF(NEXT(JPTR) .NE. LPAR) GO TO 40
      IF(BITGET(IDTBL(3,LOC),1,1) .NE. 0) GO TO 80
      I=0
   35 I=I+1
      CALL GNLE                                                         COM   41
      IF(JTYP .NE. 5) GO TO 60                                          COM   42
      IDIM(I)=N2                                                        COM   43
      ICMSIZ=ICMSIZ+N2                                                  COM   44
      IF(N2 .GT. (2**17-1)) CALL ERROR(8,KDM1,KDM2)
      IF(N2 .LE. 0) CALL ERROR(8,KDM1,KDM2)
      IF(NEXT(JPTR) .EQ. COMMA) GO TO 35                                COM   47
      IF(A(JPTR-1) .NE. RPAR) GO TO 60                                  COM   48
      K=NEXT(JPTR)                                                      COM   49
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,1)                             COM   50
      IF(I .GT. 3) GO TO 60                                             COM   51
```

```
56•          IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),I,7)                    COM   52
57•          GO TO (34,32,30),I                                       COM   53
58•       30 IDTBL(4,LOC)=BITPUT(IDTBL(4,LOC),IDIM(3),36)             COM   54
59•       32 IDTBL(4,LOC)=BITPUT(IDTBL(4,LOC),IDIM(2),18)             COM   55
60•       34 IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),IDIM(1),36)             COM   56
61•       40 IF(IDTBL(5,ICMLOC) .EQ. 0) GO TO 45                      COM   59
62•          IDTBL(5,LSTLOC)=LOC                                      COM   60
63•          GO TO 47                                                 COM   61
64•       45 IDTBL(5,ICMLOC)=LOC                                      COM   62
65•       47 IDTBL(4,ICMLOC)=IDTBL(4,ICMLOC)+ICMSIZ                   COM   63
66•          IDTBL(6,ICMLOC)=LOC                                      COM   64
67•          IDTBL(6,LOC)=ICMLOC                                      COM   65
68•          LSTLOC=LOC                                               COM   66
69•          IF(A(JPTR-1) .EQ. COMMA) GO TO 27                        COM   67
70•          IF(A(JPTR-1) .NE. SLASH) GO TO 50                        COM   68
71•          IDTBL(5,LOC)=IDTBL(5,ICMLOC)                             COM   69
72•          GO TO 15                                                 COM   70
73•       50 IF(NEXT(JPTR) .NE. BLANK) GO TO 60                       COM   71
74•          IDTBL(5,LOC)=IDTBL(5,ICMLOC)                             COM   72
75•          RETURN                                                   COM   73
76•       60 CALL ERROR(7,KDM1,KDM2)
77•          RETURN                                                   COM   75
78•       80 CALL ERROR(14,NXTID,KDM2)
79•          RETURN
80•          END                                                      COM   76
```

```
      SUBROUTINE COMCHK
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),
     • JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/LIST/NLIST,NINTFC,ISUBLT(3,200),INTFAC(500)
      COMMON/GLOBAL/NBLK,NREF,NSUBS,BLKTBL(200),EXTTBL(100),ISUBS(100)
      INTEGER BITGET,CMBLK(2,20),TP,SZ,PREVTP,BLKTBL
      INTEGER SESCOM(13),SESERR
      DIMENSION ITPS(6),IORD(6)
      DATA IORD/1,2,5,4,3,6/
      DATA SESCOM/4HCASE,3HINA,3HINB,3HINC,3HIOX,5HNPAGX,4HLINX,3HIOY,
     $ 5HNPAGY,4HLINY,3HIOZ,5HNPAGZ,4HLINZ/
      SESERR=0
      NSES=0
      ICTGR2=0
      IBLK=INITID(3)
      MODCLS=0
      LC2=BITGET(IDTBL(3,1),36,9)
      IF(LC2 .EQ. 0 .OR. IBLKDT .EQ. 1) GO TO 1
      MODCLS=BITGET(ISUBLT(2,LC2),10,4)
    1 IF(IBLK .EQ. 0) GO TO 120
      IF(IDTBL(1,IBLK) .EQ. 1H ) GO TO 3
      LISTLC=BITGET(IDTBL(3,IBLK),36,9)
      KLAS=BITGET(ISUBLT(2,LISTLC),10,4)
      ISZ=BITGET(ISUBLT(2,LISTLC),36,15)
      IF(IDTBL(4,IBLK) .NE. ISZ) GO TO 70
      GO TO 5
    3 IF(MODCLS .NE. 1 .AND. MODCLS .NE. 2) GO TO 5
      IBNKSZ=BITGET(ISUBLT(2,LC2),36,15)
      IF(IDTBL(4,IBLK) .NE. IBNKSZ) CALL ERROR(58,2H//,KDM2)
    5 NBLOC=0
      ISUM=0
      NTP=0
      TP=0
      ICOMST=IDTBL(5,IBLK)
      LOC=ICOMST
   10 PREVTP=TP
      IF(BITGET(IDTBL(3,LOC),11,1) .EQ. 1) GO TO 15
      TP=1
      IFST=BITGET(IDTBL(1,LOC),6,6)
      IF(IFST .LE. 19 .AND. IFST .GE. 14) TP=4
      GO TO 18
   15 TP=BITGET(IDTBL(3,LOC),10,3)
   18 SZ=1
      NDIM=BITGET(IDTBL(3,LOC),7,6)
      IF(NDIM .EQ. 0) GO TO 22
      DO 20 I=1,NDIM
      N#=3+(I/2)
      ICOL=18*(MOD(I,2)+1)
   20 SZ=SZ*BITGET(IDTBL(N#,LOC),ICOL,17)
   22 IF(TP .NE. 2 .AND. TP .NE. 3) GO TO 25
      IF(MOD(ISUM,2) .NE. 0)CALL ERROR(69,IDTBL(1,LOC),IDTBL(1,IBLK))
      ISUM=ISUM+SZ
   25 ISUM=ISUM+SZ
      IF(IBLKDT .EQ. 1 .AND. IORD(PREVTP+1) .GT. IORD(TP+1))
     $ CALL ERROR(65,IDTBL(1,IBLK),KDM2)
      IF(KLAS .EQ. 10 .OR. IDTBL(1,IBLK) .EQ. 1H ) GO TO 38
      IF(KLAS .EQ. 9) GO TO 35
      IF(KLAS .EQ. 7) GO TO 40
      ICTGR2=1
      IF(TP .EQ. PREVTP) GO TO 35
      IF(PREVTP .EQ. 0) GO TO 32
      DO 30 I=1,NTP
      IF(TP .EQ. ITPS(I)) GO TO 110
   30 CONTINUE
   32 NTP=NTP+1
      ITPS(NTP)=TP
   35 IF(IBLKDT .EQ. 1) GO TO 38
```

```
      IF(BITGET(IDTBL(3,LOC),19,1) .EQ. 0) CALL ERROR(75,IDTBL(1,LOC),
     $ KDM2)
   38 LOC=IDTBL(5,LOC)
      IF(LOC .NE. ICOMST) GO TO 10
      GO TO 65
   40 IF(TP .EQ. PREVTP) GO TO 45
      NBLOC=NBLOC+1
      CMBLK(1,NBLOC)=TP
      CMBLK(2,NBLOC)=0
   45 CMBLK(2,NBLOC)=CMBLK(2,NBLOC)+SZ
      IF(IDTBL(1,IBLK) .NE. 6HSESCOM) GO TO 85
      NSES=NSES+1
      IF(NSES .GT. 13) GO TO 80
      IF(IDTBL(1,LOC) .EQ. SESCOM(NSES)) GO TO 85
   80 SESERR=1
   85 LOC=IDTBL(5,LOC)
      IF(LOC .NE. ICOMST) GO TO 10
      IPTR=ISUBLT(3,LISTLC)
      NDPTR=IPTR+(NBLOC-1)/2
      KOUNT=0
      NGRP=NBLOC
      DO 50 I=IPTR,NDPTR
      ICOL=0
      DO 50 J=1,2
      KOUNT=KOUNT+1
      IF(KOUNT .GT. NBLOC) GO TO 65
      ICOL=ICOL+15
      SZ=BITGET(INTFAC(I),ICOL,15)
      ICOL=ICOL+3
C** GET GROUP TYPE
      TP=BITGET(INTFAC(I),ICOL,3)
      IF(TP .NE. 0) GO TO 48
      CMBLK(2,KOUNT)=CMBLK(2,KOUNT)-SZ
      IF(CMBLK(2,KOUNT) .EQ. 0) GO TO 50
      IF(CMBLK(2,KOUNT) .LT. 0) GO TO 90
      KOUNT=KOUNT-1
      NGRP=NGRP+1
      GO TO 50
C** CHECK INTERFACE DEFINITION FOR SIZE AND TYPE
   48 IF(CMBLK(1,KOUNT) .NE. TP .OR. CMBLK(2,KOUNT) .NE. SZ) GO TO 90
   50 CONTINUE
      IF(NGRP .NE. BITGET(ISUBLT(2,LISTLC),6,6)) GO TO 90
   65 IBLK=IDTBL(2,IBLK)
      GO TO 1
   70 CALL ERROR(58,IDTBL(1,IBLK),KDM2)
      GO TO 65
   90 CALL ERROR(57,IDTBL(1,IBLK),KDM2)
      GO TO 65
  110 CALL ERROR(63,IDTBL(1,IBLK),KDM2)
      GO TO 65
C** CHECK THAT COMMON BLOCK 'SESCOM' IS WELL DEFINED
  120 IF(ICTGR2 .EQ. 0 .AND. (MODCLS .EQ. 1 .OR. MODCLS .EQ. 2))
     $ CALL ERROR(73,KDM1,KDM2)
      IF(NSES .EQ. 0) GO TO 130
      IF(SESERR .EQ. 1 .OR. NSES .LT. 13) CALL ERROR(48,KDM1,KDM2)
      RETURN
  130 CALL ERROR(66,KDM1,KDM2)
      RETURN
      END
```

```
      SUBROUTINE COMEXT                                                 COMEXT 2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      INTEGER BITGET                                                    COMEXT 4
      ICOMLC=0                                                          COMEXT 5
      ICOMNM=IDTBL(6,LOC)                                               COMEXT 6
      ICOMST=IDTBL(5,ICOMNM)                                            COMEXT 7
      ICOMND=IDTBL(6,ICOMNM)                                            COMEXT 8
      ICOMSZ=IDTBL(4,ICOMNM)                                            COMEXT 9
      NXTLOC=ICOMND                                                     COMEXT10
      DO 20 I=1,ICOMSZ                                                  COMEXT11
      NXTLOC=IDTBL(5,NXTLOC)                                            COMEXT12
      MUL=1
      ISZ=1
      ITP=BITGET(IDTBL(3,NXTLOC),10,3)
      IF(ITP .EQ. 2 .OR. ITP .EQ. 3) MUL=2
      IF(LOC .EQ. NXTLOC) GO TO 25                                      COMEXT13
      IF(BITGET(IDTBL(3,NXTLOC),1,1) .EQ. 1) GO TO 10                   COMEXT14
      GO TO 20                                                          COMEXT16
   10 NDIM=BITGET(IDTBL(3,NXTLOC),7,6)                                  COMEXT17
      DO 15 J=1,NDIM                                                    COMEXT19
      IWRD=3+J/2                                                        COMEXT20
      IPOS=MOD(J,2)*18+18
      ISZ=ISZ*BITGET(IDTBL(IWRD,NXTLOC),IPOS,18)                        COMEXT22
   15 CONTINUE                                                          COMEXT23
   20 ICOMLC=ICOMLC+ISZ*MUL
   25 ILFT=ICOMLC                                                       COMEXT26
      IRHT=ICOMSZ-ICOMLC                                                COMEXT27
      NXTLOC=LOC                                                        COMEXT28
      IOFFST=IDTBL(8,NXTLOC)                                            COMEXT29
   30 NXTLOC=IDTBL(7,NXTLOC)                                            COMEXT30
      IF(NXTLOC .EQ. LOC) RETURN                                        COMEXT31
      IOFF2=IDTBL(8,NXTLOC)                                             COMEXT32
      IF((IOFFST-IOFF2) .GT. ILFT) GO TO 50                             COMEXT33
      ITP=BITGET(IDTBL(3,NXTLOC),10,3)
      ISZ=1                                                             COMEXT34
      IF(ITP .NE. 2 .AND. ITP .NE. 3) GO TO 32
      IF(MOD((ILFT-IOFFST+IOFF2),2) .NE. 0)
     $ CALL ERROR(64,IDTBL(1,NXTLOC),IDTBL(1,ICOMNM))
      ISZ=2
   32 IF(BITGET(IDTBL(3,NXTLOC),1,1) .NE. 1) GO TO 40
      NDIM=BITGET(IDTBL(3,NXTLOC),7,6)                                  COMEXT36
      DO 35 I=1,NDIM                                                    COMEXT37
      IWRD=3+I/2                                                        COMEXT38
      IPOS=MOD(I,2)*18+18
      ISUB=BITGET(IDTBL(IWRD,NXTLOC),IPOS,18)                           COMEXT40
      ISZ=ISZ*ISUB                                                      COMEXT41
   35 CONTINUE                                                          COMEXT42
   40 IF((ISZ-(IOFFST-IOFF2)) .GT. IRHT) GO TO 50                       COMEXT43
      GO TO 30                                                          COMEXT44
   50 CALL ERROR(47,KDM1,KDM2)
      RETURN                                                            COMEXT46
      END                                                               COMEXT47
```

```
      SUBROUTINE COMSCH                                                 COMSCH 2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     • JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NATID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      J=INITID(3)                                                       COMSCH 4
      IF(J .EQ. 0) GO TO 15                                             COMSCH 5
      DO 10 I=1,NID                                                     COMSCH 6
      IF(IDTBL(1,J) .NE. NXTID) GO TO 5                                 COMSCH 7
      ISRCH(3)=1                                                        COMSCH 8
      LOC=J                                                             COMSCH 9
      RETURN                                                            COMSCH10
    5 J=IDTBL(2,J)                                                      COMSCH11
      IF(J .EQ. 0) GO TO 15                                             COMSCH12
   10 CONTINUE                                                          COMSCH13
   15 ISRCH(3) =0                                                       COMSCH14
      RETURN                                                            COMSCH15
      END                                                               COMSCH16
```

```
      SUBROUTINE CTGOTO                                                 CTGOTO 2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     • JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      COMMON/LABELS/STATRA(2,200),NLABEL                                CTGOTO 4
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      DIMENSION IALPH(4)                                                CTGOTO 6
      INTEGER STATRA,A,BLANK,RPAR,COMMA                                 CTGOTO 7
      INTEGER BITPUT,BITGET                                             CTGOTO 8
      DATA BLANK/1H /,COMMA/1H,/,LPAR/1H(/,RPAR/1H)/                    CTGOTO 9
      DATA (IALPH(I),I=1,4)/1HG,1HO,1HT,1HO/                            CTGOTO10
      DO 5 I=1,4                                                        CTGOTO11
      IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 30                             CTGOTO12
    5 CONTINUE                                                          CTGOTO13
      IF(NEXT(JPTR) .NE. LPAR) GO TO 30                                 CTGOTO14
      NBLOCK=NBLOCK+1                                                   CTGOTO15
      JBLOCK=NBLOCK                                                     CTGOTO16
      NBRNCH=0                                                          CTGOTO17
   10 CALL GNLE                                                         CTGOTO18
      IF(JTYP .NE. 5) GO TO 30                                          CTGOTO19
      CALL STSRCH                                                       CTGOTO20
      STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,12)                          CTGOTO21
      IF(NBRNCH .EQ. 0) GO TO 15                                        CTGOTO22
      DO 12 I=1,NBRNCH                                                  CTGOTO23
      IF(LOC .EQ. IBLOCK(NBLOCK-I+1)) GO TO 17                          CTGOTO24
   12 CONTINUE                                                          CTGOTO25
   15 NBLOCK=NBLOCK+1                                                   CTGOTO26
      IBLOCK(NBLOCK)=LOC                                                CTGOTO27
      NBRNCH=NBRNCH+1                                                   CTGOTO28
   17 IF(NEXT(JPTR) .EQ. COMMA) GO TO 10                                CTGOTO29
      IF(A(JPTR-1) .NE. RPAR) GO TO 30                                  CTGOTO30
      IF(NEXT(JPTR) .NE. COMMA) GO TO 30                                CTGOTO31
      CALL GNLE                                                         CTGOTO32
      IF(JTYP .NE. 2) GO TO 30                                          CTGOTO33
      CALL SEARCH                                                       CTGOTO34
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID,KDM2)
      IF(ISRCH(1) .EQ. 1) GO TO 20                                      CTGOTO36
      IDTYP=1                                                           CTGOTO37
      CALL STORE                                                        CTGOTO38
      LOC=NID                                                           CTGOTO39
   20 CALL IMPTYP                                                       CTGOTO40
      IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4) CALL ERROR(39,NXTID,KDM2)
      IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 1) CALL ERROR(19,NXTID,KDM2)
      IF(NEXT(JPTR) .NE. BLANK) GO TO 30                                CTGOTO43
      IBLOCK(JBLOCK)=2000+LOC                                           CTGOTO44
      NB=1                                                              CTGOTO45
      RETURN                                                            CTGOTO46
   30 CALL ERROR(7,KDM1,KDM2)
      RETURN                                                            CTGOTO48
      END                                                               CTGOTO49
```

```
      SUBROUTINE DATA                                                    DATA   2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),   RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                   RICH   4
      DIMENSION IALPH(4)                                                 DATA   4
      INTEGER A,RPAR,COMMA,SLASH,BLANK,ASTRIK,PLUS                       DATA   5
      INTEGER BITPUT,BITGET                                              DATA   6
      DATA LPAR/1H(/,RPAR/1H)/,COMMA/1H,/,SLASH/1H//,BLANK/1H /,         DATA   7
     1   ASTRIK/1H*/,PLUS/1H+/,MINUS/1H-/                                DATA   8
      DATA (IALPH(I),I=1,4)/1HD,1HA,1HT,1HA/                             DATA   9
      DO 5 I=1,4                                                         DATA  10
      IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 60                              DATA  11
    5 CONTINUE                                                           DATA  12
    6 LST1SZ=0                                                           DATA  13
      LST2SZ=0                                                           DATA  14
    8 ISZ=1
      CALL GNLE
      IF(JTYP .NE. 2) GO TO 60                                           DATA  16
      CALL SEARCH                                                        DATA  17
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID,KDM2)
      IF(ISRCH(1) .EQ. 1) GO TO 9                                        DATA  19
      IDTYP=1                                                            DATA  20
      CALL STORE                                                         DATA  21
      LOC=NID                                                            DATA  22
    9 IF(BITGET(IDTBL(3,LOC),12,1) .EQ. 1) CALL ERROR(30,NXTID,KDM2)
      CALL IMPTYP                                                        DATA  24
      IF(BITGET(IDTBL(3,LOC),16,1) .EQ. 0) GOTO 10                       DATA  25
      ICOMLC=IDTBL(6,LOC)                                                DATA  27
      IF(IBLKDT .EQ. 0 .OR. IDTBL(1,ICOMLC) .EQ. BLANK)
     $ CALL ERROR(28,NXTID,KDM2)
   10 IF(NEXT(JPTR) .NE. LPAR) GO TO 25                                  DATA  29
      IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 0) GO TO 90                       DATA  31
      NDIM=BITGET(IDTBL(3,LOC),7,6)                                      DATA  32
      I=0
   15 I=I+1
      CALL GNLE                                                          DATA  34
      IF(JTYP .NE. 5) GO TO 60                                           DATA  35
      IF(N2 .LE. 0) CALL ERROR(8,KDM1,KDM2)
      IWRD=3+I/2                                                         DATA  37
      IPOS=18*MOD(I,2)+18
      IF(N2 .GT. BITGET(IDTBL(IWRD,LOC),IPOS,17))CALL ERROR(18,KD1,KD2)
      IF(NEXT(JPTR) .EQ. COMMA) GO TO 15                                 DATA  40
      IF(I .NE. NDIM) GO TO 80
      IF(A(JPTR-1) .NE. RPAR) GO TO 80                                   DATA  42
      GO TO 35                                                           DATA  43
   25 JPTR=JPTR-1
      IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 0) GO TO 35
      NDIM=BITGET(IDTBL(3,LOC),7,6)
      DO 30 I=1,NDIM
      IWRD=3+I/2
      IPOS=18*MOD(I,2)+18
   30 ISZ=ISZ*BITGET(IDTBL(IWRD,LOC),IPOS,17)
      IF(BITGET(IDTBL(3,LOC),14,1) .EQ. 1) CALL ERROR(29,IDTBL(1,LOC),K)
   35 LST1SZ=LST1SZ+ISZ
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,14)
      IF(NEXT(JPTR) .EQ. COMMA) GO TO 8                                  DATA  50
      IF(A(JPTR-1) .NE. SLASH) GO TO 60                                  DATA  51
```

```
      40 NRPEAT=1                                                        DATA  52
         CALL GNLE                                                       DATA  53
         IF(JTYP .EQ. 3) GO TO 47                                        DATA  54
         IF(JTYP .NE. 5) GO  TO 45                                       DATA  55
         IF(NEXT(JPTR) .NE. ASTRIK) GO TO 50                             DATA  56
         NRPEAT=N2                                                       DATA  57
         CALL GNLE                                                       DATA  58
      45 IF(A(JPTR-1) .NE. PLUS .AND. A(JPTR-1) .NE. MINUS) GO TO 47     DATA  59
         CALL GNLE                                                       DATA  60
      47 KK=NEXT(JPTR)                                                   DATA  61
      50 IF(JTYP .GE. 3.AND. JTYP .LE. 6) GO TO 55                       DATA  62
         IF(JTYP .EQ. 7 .AND. LOGID .EQ. 10) GO TO 55                    DATA  63
         IF(JTYP .EQ. 7 .AND. LOGID .EQ. 11) GO TO 55                    DATA  64
         GO TO 70                                                        DATA  65
      55 LST2SZ=LST2SZ+NRPEAT                                            DATA  66
         IF(A(JPTR-1) .EQ. COMMA) GO TO 40                               DATA  67
         IF(A(JPTR-1) .NE. SLASH) GO TO 60                               DATA  68
         IF(LST1SZ .NE. LST2SZ) CALL ERROR(31,KDM1,KDM2)
         IF(NEXT(JPTR) .EQ. COMMA) GO TO 6                               DATA  70
         IF(A(JPTR-1) .NE. BLANK) GO TO 60                               DATA  71
         RETURN                                                          DATA  72
      60 CALL ERROR(7,KDM1,KDM2)
         RETURN                                                          DATA  74
      70 CALL ERROR(23,KDM1,KDM2)
         RETURN                                                          DATA  76
      80 CALL ERROR(19,KDM1,KDM2)
         RETURN                                                          DATA  78
      90 CALL ERROR(13,IDTBL(1,LOC),KDM2)
         RETURN                                                          DATA  80
         END                                                             DATA  81
```

```
      SUBROUTINE DESCRP                                                 DESCRP 2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      COMMON/FORMAT/IDESST,IDESND,IGPST,IGPND,IGRP,SEPST,SEPND,
     1 DIR,ICOM,ISEP                                                    DESCRP 5
      DIMENSION FORMT(7)                                                DESCRP 6
      INTEGER A,FORMT,DECPT,BLANK,PEE,EX,AICH                           DESCRP 7
      DATA (FORMT(I),I=1,7)/1HF,1HE,1HG,1HD,1HI,1HL,1HA/                DESCRP 8
      DATA DECPT/1H./,BLANK/1H /,PEE/1HP/,EX/1HX/,AICH/1HH/,MINUS/1H-/  DESCRP 9
      ISCLFC=0                                                          DESCRP10
      INT=0                                                             DESCRP11
      IMINUS=0                                                          DESCRP12
      IDES=1                                                            DESCRP13
      IF(NEXT(IDESST) .NE. MINUS) GO TO 5                               DESCRP14
      IMINUS=1                                                          DESCRP15
      GO TO 6                                                           DESCRP16
    5 JPTR=IDESST                                                       DESCRP17
    6 CONTINUE                                                          DESCRP18
      CALL GNLE
      IF(JTYP .EQ. 3) GO TO 80
      IF(JTYP .EQ. 5) GO TO 10
      IF(JTYP .NE. 2) GO TO 15
      IF(NXTID .EQ. PEE .AND. ISCLFC .EQ. 0) GO TO 20
      IF(INT .EQ. 1 .AND. N2 .LT. 1) GO TO 15
      IF(ISCLFC .EQ. 0 .AND. IMINUS .EQ. 1) GO TO 15
      GO TO 25
   10 INT=1                                                             DESCRP24
      GO TO 6                                                           DESCRP25
   15 IDES=0                                                            DESCRP26
      RETURN                                                            DESCRP27
   20 IF(INT .EQ. 0) GO TO 15
      ISCLFC=1
      INT=0
      GO TO 6
   25 DO 30 I=1,7
      IF(NXTID .EQ. FORMT(I)) GO TO 45
   30 CONTINUE                                                          DESCRP36
      IF(NXTID .NE. EX .OR. ISCLFC .EQ. 1 .OR. INT .EQ. 0) GO TO 15
      GO TO 80
   45 CALL GNLE                                                         DESCRP41
      IF(JTYP .NE. 5) GO TO 15                                          DESCRP42
      NWIDTH=N2                                                         DESCRP43
      IF(I .LE. 4) GO TO 60
      IF(ISCLFC .EQ. 1 .OR. NWIDTH .LT. 1) GO TO 15
      IF(I .EQ. 7 .AND. NWIDTH .GT. 4) GO TO 15
      IDESND=JPTR-1                                                     DESCRP47
      RETURN                                                            DESCRP48
   60 IF(NEXT(JPTR) .NE. DECPT) GO TO 15                                DESCRP49
      IF(NWIDTH .LT. 2) GO TO 15                                        DESCRP50
      CALL GNLE                                                         DESCRP51
      IF(JTYP .NE. 5) GO TO 15                                          DESCRP52
      NDCPLS=N2                                                         DESCRP53
      IDESND=JPTR-1                                                     DESCRP54
      IF(I .EQ. 1) GO TO 65                                             DESCRP55
      IF(NWIDTH .LT. (NDCPLS+6)) GO TO 15                               DESCRP56
      RETURN                                                            DESCRP57
   65 IF(NWIDTH .LT. NDCPLS) GO TO 15                                   DESCRP58
      RETURN                                                            DESCRP59
   80 IDESND=JPTR-1                                                     DESCRP60
      RETURN                                                            DESCRP61
      END                                                               DESCRP67
```

```
      SUBROUTINE DIMEN                                                  DIMEN  2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      DIMENSION IALPH(9),IDIM(3)                                        DIMEN  4
      INTEGER A,D,RPAR,COMMA                                            DIMEN  5
      INTEGER BITPUT,BITGET,COMLOC                                      DIMEN  6
      DATA (IALPH(I),I=1,9)/1HD,1HI,1HM,1HE,1HN,1HS,1HI,1HO,1HN/        DIMEN  7
      DATA LPAR/1H(/,RPAR/1H)/,COMMA/1H,/                               DIMEN  8
      DO 10 I=1,9                                                       DIMEN  9
      IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 110                            DIMEN 10
   10 CONTINUE                                                          DIMEN 11
   12 CALL GNLE                                                         DIMEN 12
      IF(JTYP .NE. 2) GO TO 110                                         DIMEN 13
      CALL SEARCH                                                       DIMEN 14
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID,KDM2)
      IF(ISRCH(1) .EQ. 1) GO TO 5                                       DIMEN 16
      IDTYP=1                                                           DIMEN 17
      CALL STORE                                                        DIMEN 18
      LOC=NID                                                           DIMEN 19
    5 IF(BITGET(IDTBL(3,LOC),1,1) .NE. 0) CALL ERROR(11,NXTID,KDM1)
      CALL IMPTYP                                                       DIMEN 21
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,1)                             DIMEN 22
      IE=LOC                                                            DIMEN 23
      IF(NEXT(JPTR) .NE. LPAR) GO TO 110                                DIMEN 24
      INCR=1                                                            DIMEN 25
      I=0
   15 I=I+1
      CALL GNLE                                                         DIMEN 27
      IF(JTYP .NE. 5) GO TO 13                                          DIMEN 28
      IDIM(I)=N2                                                        DIMEN 29
      IF(N2 .GT. (2**17-1)) CALL ERROR(8,KDM1,KDM2)
      IF(N2 .LE. 0) CALL ERROR(8,KDM1,KDM2)
      INCR=INCR*N2                                                      DIMEN 32
      GO TO 14                                                          DIMEN 33
   13 IF(JTYP .NE. 2) GO TO 110                                         DIMEN 34
      IDTYP=1                                                           DIMEN 35
      CALL SEARCH                                                       DIMEN 36
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID,KDM2)
      IF(ISRCH(1) .EQ. 1) GOTO 25                                       DIMEN 38
      IDTYP=1                                                           DIMEN 39
      CALL STORE                                                        DIMEN 40
      LOC=NID                                                           DIMEN 41
   25 IF(BITGET(IDTBL(3,LOC),12,1) .NE. 1) CALL ERROR(9,KDM1,KDM2)
      IF(BITGET(IDTBL(3,LOC),1,1) .NE. 0) GO TO 120
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,13)                            DIMEN 44
      CALL IMPTYP                                                       DIMEN 45
      IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4) CALL ERROR(9,KDM1,KDM2)
      IDIM(I)=2**17*LOC                                                 DIMEN 47
   14 IF(NEXT(JPTR) .EQ. COMMA) GO TO 15                                DIMEN 48
      IF(A(JPTR-1) .NE. RPAR) GO TO 110
      LOC=IE
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,7)                             DIMEN 53
      IF(I .GT. 3) GO TO 110                                            DIMEN 54
      GO TO (35,30,24),I                                                DIMEN 55
   24 IDTBL(4,LOC)=BITPUT(IDTBL(4,LOC),IDIM(3),36)                      DIMEN 56
   30 IDTBL(4,LOC)=BITPUT(IDTBL(4,LOC),IDIM(2),18)                      DIMEN 57
   35 IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),IDIM(1),36)                      DIMEN 58
      IF(BITGET(IDTBL(3,LOC),16,1) .NE. 1) GO TO 50                     DIMEN 59
      COMLOC=IDTBL(6,LOC)                                               DIMEN 60
      IT=1                                                              DIMEN 61
      ITP=BITGET(IDTBL(3,LOC),10,3)                                     DIMEN 62
      IF(ITP .EQ. 2 .OR. ITP .EQ. 3) IT=2
      IDTBL(4,COMLOC)=IDTBL(4,COMLOC)+IT*(INCR-1)                       DIMEN 64
   50 CONTINUE                                                          DIMEN 65
      IF(NEXT(JPTR) .EQ. COMMA) GO TO 12                                DIMEN 66
      IF(JPTR .GT. N) RETURN                                            DIMEN 67
  110 CALL ERROR(7,KDM1,KDM2)
      RETURN                                                            DIMEN 69
  120 CALL ERROR(14,NXTID,KDM2)
      RETURN
      END                                                               DIMEN 70
```

```
      SUBROUTINE DO                                                    DO     2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                 RICH   4
      COMMON/LABELS/STATRA(2,200),NLABEL                               DO     4
      COMMON/DOLOOP/ISTACK(4,50),NSTACK,ILOOP,IOVFL#
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      DIMENSION PARAM(3)                                               DO     7
      INTEGER A,BLANK,COMMA,EQUALS,DEE,OH,PARAM,STATRA                 DO     8
      INTEGER BITPUT,BITGET                                            DO     9
      DATA BLANK/1H /,COMMA/1H,/,EQUALS/1H=/,DEE/1HD/,OH/1HO/          DO     10
      IF(NEXT(JPTR) .NE. DEE) GO TO 50                                 DO     11
      IF(NEXT(JPTR) .NE. OH) GO TO 50                                  DO     12
      CALL GNLE                                                        DO     13
      IF(JTYP .NE. 5) GOTO 50                                          DO     14
      CALL STSRCH                                                      DO     15
      STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,12)                         DO     16
      STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,15)                         DO     17
      IF(IOVFL# .EQ. 1) GO TO 2
      NSTACK=NSTACK+1                                                  DO     18
      IF(NSTACK .GT. 50) GO TO 1
      ISTACK(1,NSTACK)=LOC                                             DO     19
      ISTACK(2,NSTACK)=0                                               DO     20
      ISTACK(3,NSTACK)=ILOOP                                           DO     21
      ILOOP=NSTACK                                                     DO     22
      GO TO 2
    1 IOVFL#=1
      #RITE(6,60)
   60 FORMAT(///5X,50H DO STACK OVERFLOW - DO LOOP PROCESSING TERMINATED
     *)
    2 CALL GNLE
      IF(JTYP .NE. 2) GO TO 50                                         DO     24
      CALL SEARCH                                                      DO     25
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID,KDM2)
      IF(ISRCH(1) .EQ. 1) GO TO 5                                      DO     27
      IDTYP=1                                                          DO     28
      CALL STORE                                                       DO     29
      LOC=NID                                                          DO     30
    5 CALL IMPTYP                                                      DO     31
      IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4) CALL ERROR(40,NXTID,KDM2)
      IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 1) CALL ERROR(19,NXTID,KDM2)
      IF(NEXT(JPTR) .NE. EQUALS) GO TO 50                              DO     34
      IF(IOVFL# .EQ. 1) GO TO 8
      NBLOCK=NBLOCK+1                                                  DO     35
      IBLOCK(NBLOCK)=3000+LOC                                          DO     36
      ISTACK(4,NSTACK)=LOC                                             DO     37
    8 PARAM(3)=1
      DO 30 I=1,3                                                      DO     39
      CALL GNLE                                                        DO     40
      IF(JTYP .NE. 5) GO TO 10                                         DO     41
      PARAM(I)=N2                                                      DO     42
      IF(N2 .LE. 0) CALL ERROR(41,KDM1,KDM2)
      GO TO 20                                                         DO     44
   10 IF(JTYP .NE. 2) GO TO 50                                         DO     45
      CALL SEARCH                                                      DO     46
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID,KDM2)
      IF(ISRCH(1) .EQ. 1) GO TO 15                                     DO     48
      IDTYP=1                                                          DO     49
      CALL STORE                                                       DO     50
      LOC=NID                                                          DO     51
   15 CALL IMPTYP                                                      DO     52
      IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4) CALL ERROR(40,NXTID,KDM2)
      IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 1) CALL ERROR(19,NXTID,KDM2)
      NBLOCK=NBLOCK+1                                                  DO     55
      IBLOCK(NBLOCK)=7000+LOC
```

```
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),ILOOP,36)
      PARAM(I)=0                                                        DO    57
   20 IF(I .EQ. 3) GO TO 30                                             DO    58
      IF(I .EQ. 1) GO TO 25                                             DO    59
      IF(NEXT(JPTR) .EQ. BLANK) GO TO 35                                DO    60
      JPTR=JPTR+1                                                       DO    61
   25 IF(NEXT(JPTR) .NE. COMMA) GO TO 50                                DO    62
   30 CONTINUE                                                          DO    63
      IF(NEXT(JPTR) .NE. BLANK) GO TO 50                                DO    64
   35 IF(PARAM(1) .EQ. 0 .OR. PARAM(2) .EQ. 0) GO TO 40                 DO    65
      IF(PARAM(2) .LT. PARAM(1)) CALL ERROR(41,KDM1,KDM2)
      IF(PARAM(3) .EQ. 0) GO TO 40                                      DO    67
      IF((PARAM(2)+PARAM(3)-1) .GT. (2**17-1)) CALL ERROR(41,KDM1,KDM2)
   40 NBLOCK=NBLOCK+1                                                   DO    69
      IBLOCK(NBLOCK)=998                                                DO    70
      NBRNCH=1                                                          DO    71
      NB=1                                                              DO    72
      RETURN                                                            DO    73
   50 CALL ERROR(7,KDM1,KDM2)
      RETURN                                                            DO    75
      END                                                               DO    76
```

```
      SUBROUTINE EQUIV                                                  EQUIV  2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      DIMENSION IALPH(11),IDIM(3),ISUB(3)                               EQUIV  4
      INTEGER BITPUT,BITGET                                             EQUIV  5
      INTEGER A,RPAR,COMMA,BOFFST,BLANK                                 EQUIV  6
      DATA (IALPH(I),I=1,11)/1HE,1HQ,1HU,1HI,1HV,1HA,1HL,1HE,1HN,1HC,  EQUIV  7
     1 1HE/                                                             EQUIV  8
      DATA LPAR/1H(/,RPAR/1H)/,COMMA/1H,/,BLANK/1H /                    EQUIV  9
      DO 5 I=1,11                                                       EQUIV 10
      IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 130                            EQUIV 11
    5 CONTINUE                                                          EQUIV 12
    8 IF(NEXT(JPTR) .NE. LPAR) GO TO 130                                EQUIV 13
      LSTLOC=0                                                          EQUIV 14
      BOFFST=0                                                          EQUIV 15
      J=0                                                               EQUIV 16
  120 J=J+1                                                             EQUIV 17
      CALL GNLE                                                         EQUIV 18
      ILOC=1                                                            EQUIV 19
      IF(JTYP .NE. 2) GO TO 130                                         EQUIV 20
      CALL SEARCH                                                       EQUIV 21
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID,KDM2)                     EQUIV 22
      IF(ISRCH(1) .EQ. 1) GO TO 9                                       EQUIV 23
      IDTYP=1                                                           EQUIV 24
      CALL STORE                                                        EQUIV 25
      LOC=NID                                                           EQUIV 26
    9 CALL IMPTYP                                                       EQUIV 27
      IF(BITGET(IDTBL(3,LOC),12,1) .EQ. 1) CALL ERROR(20,NXTID,KDM2)    EQUIV 28
      IF(NEXT(JPTR) .NE. LPAR) GO TO 30                                 EQUIV 29
      IF(BITGET(IDTBL(3,LOC),1,1) .NE. 1) GO TO 150                     EQUIV 30
      NDIM=BITGET(IDTBL(3,LOC),7,6)                                     EQUIV 31
      DO 10 I=1,NDIM                                                    EQUIV 32
      CALL GNLE                                                         EQUIV 33
      IF(JTYP .NE. 5) GO TO 130                                         EQUIV 34
      IDIM(I)=N2                                                        EQUIV 35
      IF(N2 .LE. 0) CALL ERROR(8,KDM1,KDM2)                             EQUIV 36
      IWRD=3+I/2                                                        EQUIV 37
      IPOS=18*MOD(I,2)+18                                               EQUIV 38
      ISUB(I)=BITGET(IDTBL(IWRD,LOC),IPOS,17)                           EQUIV 39
      IF(N2 .GT. ISUB(I)) CALL ERROR(18,KDM1,KDM2)                      EQUIV 40
      IF(NEXT(JPTR) .EQ. COMMA) GO TO 10                                EQUIV 41
      IF(A(JPTR-1) .NE. RPAR) GO TO 130                                 EQUIV 42
      GO TO 15                                                          EQUIV 43
   10 CONTINUE                                                          EQUIV 44
      GO TO 140                                                         EQUIV 45
   15 NDIM=I                                                            EQUIV 46
      ILOC=IDIM(1)                                                      EQUIV 47
      IF(NDIM .EQ. 1) GO TO 25                                          EQUIV 48
      ILOC=ILOC+(IDIM(2)-1)*ISUB(1)                                     EQUIV 49
      IF(NDIM .EQ. 2) GO TO 25                                          EQUIV 50
      ILOC=ILOC+(IDIM(3)-1)*ISUB(1)*ISUB(2)                             EQUIV 51
   25 IT=BITGET(IDTBL(3,LOC),10,3)                                      EQUIV 52
      IF(IT .EQ. 2 .OR. IT .EQ. 3) ILOC=2*ILOC                          EQUIV 53
      IOFFST=1-ILOC-BOFFST                                              EQUIV 54
      GO TO 45                                                          EQUIV 55
   30 IF(BITGET(IDTBL(3,LOC),1,1) .NE. 0) CALL ERROR(19,NXTID,KDM2)     EQUIV 56
      IOFFST=BOFFST                                                     EQUIV 57
      JPTR=JPTR-1                                                       EQUIV 58
   45 IF(BITGET(IDTBL(3,LOC),17,1) .EQ. 1) GO TO 57                     EQUIV 59
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,17)                            EQUIV 60
      IF(LSTLOC .EQ. 0) GO TO 50                                        EQUIV 61
      IDTBL(7,LSTLOC)=LOC                                               EQUIV 62
      GO TO 55                                                          EQUIV 63
   50 IFSTLC=LOC                                                        EQUIV 64
   55 IDTBL(8,LOC)=IOFFST                                               EQUIV 65
      LSTLOC=LOC                                                        EQUIV 66
      IF(J .NE. 1) GO TO 100                                            EQUIV 67
```

```
      GO TO 98                                                          EQUIV 67
   57 LOC3=LOC                                                          EQUIV 68
   58 LOC3=IDTBL(7,LOC3)                                                EQUIV 69
      IF(LOC3 .EQ. 0) GO TO 59                                          EQUIV 70
      IF(LOC3 .EQ. LOC) GO TO 60                                        EQUIV 71
      GO TO 58                                                          EQUIV 72
   59 JLOC=ILOC+IDTBL(8,LOC)                                            EQUIV 73
      IF(JLOC .NE. IDIS) CALL ERROR(20,IDTBL(1,LOC),KDM2)
      GO TO 100                                                         EQUIV 75
   60 IF(LSTLOC .NE. 0) GO TO 63                                        EQUIV 76
      IFSTLC=LOC                                                        EQUIV 77
      GO TO 65                                                          EQUIV 78
   63 IDTBL(7,LSTLOC)=LOC                                               EQUIV 79
   65 LOC2=LOC                                                          EQUIV 80
   70 NXTLOC=IDTBL(7,LOC2)
      IF(NXTLOC .EQ. LOC) GO TO 75                                      EQUIV 83
      LOC2=NXTLOC                                                       EQUIV 84
      GO TO 70
   75 IDTBL(7,LOC2)=0                                                   EQUIV 86
      LSTLOC=LOC2                                                       EQUIV 87
      IOFFDF=IOFFST-IDTBL(8,LOC)                                        EQUIV 88
      IF(IOFFDF) 80,98,90                                               EQUIV 89
   80 LOC2=LOC                                                          EQUIV 90
   85 IDTBL(8,LOC2)=IDTBL(8,LOC2)+IOFFDF
      LOC2=IDTBL(7,LOC2)                                                EQUIV 93
      IF(LOC2 .EQ. 0) GO TO 98                                          EQUIV 94
      GO TO 85
   90 LOC2=IFSTLC                                                       EQUIV 96
   95 IF(LOC2 .EQ. LOC) GO TO 97
      IDTBL(8,LOC2)=IDTBL(8,LOC2)-IOFFDF                                EQUIV 99
      LOC2=IDTBL(7,LOC2)                                                EQUIV100
      GO TO 95
   97 BOFFST=BOFFST-IOFFDF                                              EQUIV102
   98 IDIS=ILOC+IDTBL(8,LOC)                                            EQUIV103
  100 IF(NEXT(JPTR) .EQ. COMMA) GO TO 120                               EQUIV104
      IF(A(JPTR-1) .NE. RPAR) GO TO 130                                 EQUIV105
      IF(J .EQ. 1) GO TO 130                                            EQUIV106
      IDTBL(7,LSTLOC)=IFSTLC                                            EQUIV107
      JK=0                                                              EQUIV108
      LOC3=IFSTLC                                                       EQUIV109
      I=0
  110 I=I+1
      LOC3=IDTBL(7,LOC3)                                                EQUIV111
      IF(BITGET(IDTBL(3,LOC3),16,1) .EQ. 0) GO TO 105                   EQUIV112
      JK=JK+1                                                           EQUIV113
      IF(JK .GT. 1) CALL ERROR(21,IDTBL(1,LOC3),IDTBL(1,LOC1))
      LOC1=LOC3
      LOC=LOC3                                                          EQUIV115
      CALL COMEXT                                                       EQUIV116
  105 IF(LOC3 .NE. IFSTLC) GO TO 110
      IF(NEXT(JPTR) .EQ. BLANK) RETURN                                  EQUIV120
      IF(A(JPTR-1) .EQ. COMMA) GO TO 8
  130 CALL ERROR(7,KDM1,KDM2)
      RETURN                                                            EQUIV125
  140 CALL ERROR(19,KDM1,KDM2)
      RETURN                                                            EQUIV127
  150 CALL ERROR(13,NXTID,KDM2)
      RETURN                                                            EQUIV129
      END                                                               EQUIV130
```

```
      SUBROUTINE ERROR(IERROR,INUM,INUM2)
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/FLOW/IFL,IRP
      WRITE(6,1)                                                        ERROR  3
    1 FORMAT(1X,100H**********************************************************ERROR  4
     1**********************************************)                   ERROR  5
      GO TO (5,15,25,35,45,55,65,75,85,95,105,115,125,135,145,155,165,  ERROR  6
     A175,185,195,205,215,225,235,245,255,265,275,285,295,305,315,325,  ERROR  7
     * 335,345,355,365,375,385,395,405,415,425,435,445,455,465,475,485,
     * 495,505,515,525,535,545,555,565,575,585,595,605,615,625,635,645,
     $ 655,665,675,685,695,705,715,725,735,745,755,765,775,785,795,805,
     $815,825,835,845,855,865,875,885,895,905,915,925,935,945),IERROR
      WRITE(6, 10)                                                      ERROR 10
   10 FORMAT(6X,26H THIS STATEMENT IS ILLEGAL)                          ERROR 11
      GO TO 1000                                                        ERROR 12
   15 WRITE(6, 20)                                                      ERROR 13
   20 FORMAT(6X,31H THIS STATEMENT IS OUT OF ORDER)                     ERROR 14
      GO TO 1000                                                        ERROR 15
   25 WRITE(6, 30)                                                      ERROR 16
   30 FORMAT(6X,39H VALUE OF INTEGER CONSTANT IS TOO LARGE)             ERROR 17
      GO TO 1000                                                        ERROR 18
   35 WRITE(6, 40)                                                      ERROR 19
   40 FORMAT(6X,28H TOO MANY CONTINUATION CARDS)                        ERROR 20
      GO TO 1000                                                        ERROR 21
   45 WRITE(6, 50)                                                      ERROR 22
   50 FORMAT(6X,30H HOLLERITH STRING IS TOO LARGE)                      ERROR 23
      GO TO 1000                                                        ERROR 24
   55 WRITE(6, 60)                                                      ERROR 25
   60 FORMAT(6X,26H VARIABLE NAME IS TOO LONG)                          ERROR 26
      GO TO 1000                                                        ERROR 27
   65 WRITE(6, 70)                                                      ERROR 28
   70 FORMAT(6X,31H SYNTAX ERROR IN THIS STATEMENT)                     ERROR 29
      IF(ITYP .LE. 18 .AND. IFL .GT. 0) IFL=-1
      GO TO 1000                                                        ERROR 30
   75 WRITE(6, 80)                                                      ERROR 31
   80 FORMAT(6X,46H ARRAY DIMENSION IS OUTSIDE OF ALLOWABLE RANGE)      ERROR 32
      GO TO 1000                                                        ERROR 33
   85 WRITE(6, 90)                                                      ERROR 34
   90 FORMAT(6X,45H ILLEGAL VARIABLE DIMENSION IN THIS STATEMENT)       ERROR 35
      GO TO 1000                                                        ERROR 36
   95 WRITE(6,100) INUM
  100 FORMAT(6X,33H THE FUNCTION OR SUBROUTINE NAME ,A6,18H IS USED ILLE
```

```
     *GALLY)
      GO TO 1000
  105 WRITE(6,110) INUM
  110 FORMAT(6X,14H THE VARIABLE ,A6,32H HAS BEEN PREVIOUSLY DIMENSIONED
     *)
      GO TO 1000
  115 WRITE(6,120) INUM
  120 FORMAT(6X,14H THE VARIABLE ,A6,26H HAS BEEN PREVIOUSLY TYPED)
      GO TO 1000
  125 WRITE(6,130) INUM
  130 FORMAT(6X,14H THE VARIABLE ,A6,38H IS ILLEGALLY FOLLOWED BY A LEFT
     * PAREN)
      GO TO 1000
  135 WRITE(6,140) INUM
  140 FORMAT(6X,26H THE DIMENSIONED VARIABLE ,A6,18H IS USED ILLEGALLY)
      GO TO 1000
  145 WRITE(6,150) INUM
  150 FORMAT(6X,18H STATEMENT NUMBER ,I5,15H IS NOT DEFINED)
      IF(IFL .GT. 0) IFL=-1
      GO TO 1000
  155 WRITE(6,160) INUM
  160 FORMAT(6X,18H STATEMENT NUMBER ,I5,18H IS NOT REFERENCED)
      GO TO 1000
  165 WRITE(6,170) INUM
  170 FORMAT(6X,18H ILLEGAL VARIABLE ,A6,10H IN COMMON)
      GO TO 1000                                                    ERROR 60
  175 WRITE(6,180)                                                  ERROR 61
  180 FORMAT(6X,43H VALUE OF ARRAY SUBSCRIPT EXCEEDS DIMENSION)     ERROR 62
      GO TO 1000                                                    ERROR 63
  185 WRITE(6,190)                                                  ERROR 64
  190 FORMAT(6X,25H ERROR IN ARRAY SUBSCRIPT)                       ERROR 65
      GO TO 1000                                                    ERROR 66
  195 WRITE(6,200) INUM
  200 FORMAT(6X,18H ILLEGAL VARIABLE ,A6,16H IS EQUIVALENCED)
      GO TO 1000
  205 WRITE(6,210) INUM,INUM2
  210 FORMAT(6X,22H THE COMMON VARIABLES ,A6,5H AND ,A6,17H ARE EQUIVALE
     *NCED)
      GO TO 1000                                                    ERROR 72
  215 WRITE(6,220)                                                  ERROR 73
  220 FORMAT(6X,19H ILLEGAL I/O DEVICE)                             ERROR 74
      GO TO 1000                                                    ERROR 75
  225 WRITE(6,230)                                                  ERROR 76
  230 FORMAT(6X,37H ILLEGAL CHARACTER IN THIS EXPRESSION)           ERROR 77
      GO TO 1000                                                    ERROR 78
  235 WRITE(6,240) INUM
  240 FORMAT(6X,25H ILLEGAL SUBROUTINE NAME ,A6)
      GO TO 1000
  245 WRITE(6,250)
  250 FORMAT(6X,50H SUBROUTINE TABLE OVERFLOW - PROCESSING TERMINATED)
      GO TO 1000
  255 WRITE(6,260) INUM
  260 FORMAT(6X,77H INCORRECT NUMBER OF ARGUMENTS IN CALLING SEQUENCE OF
     $ FUNCTION OR SUBROUTINE ,A6)
      GO TO 1000                                                    ERROR 87
  265 WRITE(6,270)                                                  ERROR 88
  270 FORMAT(6X,19H ILLEGAL ASSIGNMENT)                             ERROR 89
      GO TO 1000                                                    ERROR 90
```

```
      275 WRITE(6,280) INUM
      280 FORMAT(6X,14H THE VARIABLE ,A6,42H APPEARS IN A DATA STATEMENT AND
     • IN COMMON)
          GO TO 1000
      285 WRITE(6,290) INUM
      290 FORMAT(6X,14H THE VARIABLE ,A6,44H HAS PREVIOUSLY APPEARED IN A DA
     •TA STATEMENT)
          GO TO 1000
      295 WRITE(6,300) INUM
      300 FORMAT(6X,22H THE FORMAL PARAMETER ,A6,31H APPEARS IN THIS DATA ST
     •ATEMENT)
          GO TO 1000                                                    ERROR 99
      305 WRITE(6,310)                                                  ERROR100
      310 FORMAT(6X,24H LIST SIZES DO NOT MATCH)                        ERROR101
          GO TO 1000                                                    ERROR102
      315 WRITE(6,320)                                                  ERROR103
      320 FORMAT(6X,24H ILLEGAL STATEMENT LABEL)                        ERROR104
          IF(IFL .GT. 0) IFL=-1
          GO TO 1000                                                    ERROR105
      325 WRITE(6,330)                                                  ERROR106
      330 FORMAT(6X,26H DUPLICATE STATEMENT LABEL)                      ERROR107
          IF(IFL .GT. 0) IFL=-1
          GO TO 1000                                                    ERROR108
      335 WRITE(6,340)                                                  ERROR109
      340 FORMAT(6X,34H THIS STATEMENT CAN NOT BE REACHED)              ERROR110
          GO TO 1000                                                    ERROR111
      345 WRITE(6,350)                                                  ERROR112
      350 FORMAT(6X,31H DO LOOPS ARE IMPROPERLY NESTED)                 ERROR113
          GO TO 1000                                                    ERROR114
      355 WRITE(6,360)                                                  ERROR115
      360 FORMAT(6X,32H FORMAT STATEMENT IS NOT LABELED)                ERROR116
          GO TO 1000                                                    ERROR117
      365 WRITE(6,370)                                                  ERROR118
      370 FORMAT(6X,20H ILLEGAL DO TERMINAL)                            ERROR119
          GO TO 1000                                                    ERROR120
      375 WRITE(6,380)                                                  ERROR121
      380 FORMAT(6X,37H LAST EXECUTABLE STATEMENT IS ILLEGAL)           ERROR122
          IF(IFL .GT. 0) IFL=-1
          GO TO 1000                                                    ERROR123
      385 WRITE(6,390) INUM
      390 FORMAT(6X,24H THE VARIABLE REFERENCE ,A6,18H IS NOT AN INTEGER)
          GO TO 1000
      395 WRITE(6,400) INUM
      400 FORMAT(6X,27H THE DO PARAMETER OR INDEX ,A6,18H IS NOT AN INTEGER)
          GO TO 1000                                                    ERROR129
      405 WRITE(6,410)                                                  ERROR130
      410 FORMAT(6X,52H VALUE OF DO PARAMETER IS OUTSIDE OF ALLOWABLE RANGE.)ERROR131
          GO TO 1000                                                    ERROR132
      415 WRITE(6,420)                                                  ERROR133
      420 FORMAT(6X,32H COMPLEX EXPRESSIONS ARE ILLEGAL)                ERROR134
          GO TO 1000                                                    ERROR135
      425 WRITE(6,430)                                                  ERROR136
      430 FORMAT(6X,24H ILLEGAL VARIABLE FORMAT)                        ERROR137
          GO TO 1000                                                    ERROR138
      435 WRITE(6,440)                                                  ERROR139
      440 FORMAT(6X,39H THIS STATEMENT SHOULD HAVE AN I/O LIST)         ERROR140
          GO TO 1000                                                    ERROR141
      445 WRITE(6,450)                                                  ERROR142
```

```
      450 FORMAT(6X,50H STATEMENT FOLLOWING LOGICAL EXPRESSION IS ILLEGAL)  ERROR193
          GO TO 1000                                                       ERROR194
      455 #RITE(6,460)                                                     ERROR195
      460 FORMAT(6X,44H REAL NUMBER LIES OUTSIDE OF ALLO#ABLE RANGE)       ERROR196
          GO TO 1000                                                       ERROR197
      465 #RITE(6,470)                                                     ERROR198
      470 FORMAT(6X,42H THIS EQUIVALENCE STATEMENT EXTENDS COMMON)         ERROR199
          GO TO 1000                                                       ERROR150
      475 #RITE(6,480)
      480 FORMAT(6X,40H ILLEGAL VARIABLE IN COMMON BLOCK SESCOM)
          GO TO 1000
      485 #RITE(6,490) INUM
      490 FORMAT(6X,12H SUBPROGRAM ,A6,19H HAS INCORRECT TYPE)
          GO TO 1000
      495 #RITE(6,500) INUM
      500 FORMAT(6X,23H #ARNING - ARGUMENT NO.,I3,34H MAY HAVE INCORRECT DIM
     *ENSIONALITY)
          GO TO 1000
      505 #RITE(6,510) INUM
      510 FORMAT(6X,13H ARGUMENT NO.,I3,19H HAS INCORRECT TYPE)
          GO TO 1000
      515 #RITE(6,520)
      520 FORMAT(6X,49H #ARNING - THIS MODULE IS NOT IN THE SESCOMP LIST)
          GO TO 1000
      525 #RITE(6,530) INUM
      530 FORMAT(6X,14H THE VARIABLE ,A6,29H PREVIOUSLY APPEARS IN COMMON)
          GO TO 1000
      535 #RITE(6,540) INUM
      540 FORMAT(6X,13H ARGUMENT NO.,I3,11H IS INVALID)
          GO TO 1000
      545 #RITE(6,550) INUM
      550 FORMAT(6X,13H ARGUMENT NO.,I3,29H IS DESIGNATED LOGICAL OUTPUT)
          GO TO 1000
      555 #RITE(6,560) INUM
      560 FORMAT(6X,29H ILLEGAL COMMON BLOCK NAME - ,A6)
          GO TO 1000
      565 #RITE(6,570) INUM
      570 FORMAT(6X,41H #ARNING - VARIABLE TYPE IN COMMON BLOCK ,A6,41H DOES
     $ NOT AGREE #ITH INTERFACE DEFINITION)
          GO TO 1000
      575 #RITE(6,580) INUM
      580 FORMAT(6X,14H COMMON BLOCK ,A6,19H HAS INCORRECT SIZE)
          GO TO 1000
      585 #RITE(6,590)
      590 FORMAT(6X,58H EXTERNAL REFERENCE TABLE OVERFLO# - PROCESSING TERMI
     $NATED)
          GO TO 1000
      595 #RITE(6,600)
      600 FORMAT(6X,52H COMMON BLOCK TABLE OVERFLO# - PROCESSING TERMINATED)
          GO TO 1000
      605 #RITE(6,610) INUM
      610 FORMAT(6X,29H ILLEGAL COMMON BLOCK NAME - ,A6)
          GO TO 1000
      615 #RITE(6,620) INUM
      620 FORMAT(6X,14H COMMON BLOCK ,A6,27H IS NOT IN THE SESCOMP LIST)
          GO TO 1000
      625 #RITE(6,630) INUM
      630 FORMAT(6X,25H CATEGORY 2 COMMON BLOCK ,A6,23H IS NOT GROUPED BY TY
```

```
219•          $PE)
220•           GO TO 1000
221•       635 WRITE(6,640) INUM,INUM2
222•       640 FORMAT(6X,38H DOUBLE PRECISION OR COMPLEX VARIABLE ,A6,56H DOES NO
223•          $T BEGIN ON AN EVEN LOCATION WITHIN COMMON BLOCK ,A6)
224•           GO TO 1000
225•       645 WRITE(6,650) INUM
226•       650 FORMAT(6X,26H VARIABLE IN COMMON BLOCK ,A6,16H IS OUT OF ORDER)
227•           GO TO 1000
228•       655 WRITE(6,660)
229•       660 FORMAT(6X,56H THE COMMON BLOCK SESCOM DOES NOT APPEAR IN THIS PROG
230•          $RAM)
231•           GO TO 1000
232•       665 WRITE(6,670) INUM
233•       670 FORMAT(6X,14H THE DO INDEX ,A6,13H IS REDEFINED)
234•           RETURN
235•       675 WRITE(6,680) INUM
236•       680 FORMAT(6X,24H THE VARIABLE DIMENSION ,A6,13H IS REDEFINED)
237•           RETURN
238•       685 WRITE(6,690) INUM
239•       690 FORMAT(6X,23H THE ASSIGNED VARIABLE ,A6,24H IS ILLEGALLY REFERENCE
240•          $D)
241•           RETURN
242•       695 WRITE(6,700) INUM
243•       700 FORMAT(6X,14H THE VARIABLE ,A6,30H IS REFERENCED BUT NOT DEFINED)
244•           RETURN
245•       705 WRITE(6,710) INUM
246•       710 FORMAT(6X,14H THE VARIABLE ,A6,45H IS REFERENCED ILLEGALLY BY AN A
247•          $SSIGNED GO TO)
248•           RETURN
249•       715 WRITE(6,720) INUM
250•       720 FORMAT(6X,18H THE DO PARAMETER ,A6,13H IS REDEFINED)
251•           RETURN
252•       725 WRITE(6,730)
253•       730 FORMAT(6X,49H THIS MODULE CONTAINS NO CATEGORY 2 COMMON BLOCKS)
254•           GO TO 1000
255•       735 WRITE(6,740) INUM
256•       740 FORMAT(6X,24H THE ANSI FUNCTION NAME ,A6,27H IS MISUSED IN THIS PR
257•          $OGRAM)
258•           GO TO 1000
259•       745 WRITE(6,750) INUM
260•       750 FORMAT(6X,14H THE VARIABLE ,A6,60H APPEARS IN A CATEGORY 2 OR 3 CO
261•          $MMON BLOCK BUT IS NEVER USED)
262•           GO TO 1000
263•       755 WRITE(6,760)
264•       760 FORMAT(6X,75H ARRAY SUBSCRIPT OR IMPLIED DO PARAMETER MAY LIE OUTS
265•          $IDE OF ALLOWABLE RANGE)
266•           GO TO 1000
267•       765 WRITE(6,770) INUM,INUM2
268•       770 FORMAT(6X,22H MIXED MODE COMBINING ,A6,6H WITH ,A6)
269•           GO TO 1000
270•       775 WRITE(6,780) INUM
271•       780 FORMAT(6X,33H INCORRECT EXPONENT AT CHAR. NO. ,I3)
272•           GO TO 1000
273•       785 WRITE(6,790) INUM
274•       790 FORMAT(6X,47H VAR-CONST CONFUSION IN SUBSCRIPT AT CHAR. NO. ,I3)
275•           GO TO 1000
276•       795 WRITE(6,800) INUM,INUM2
```

```
277•        800 FORMAT(6X,40H SUBSCRIPT CONSTANT OR VARIABLE OF TYPE ,A6,19H AT CH
278•           $AR. NO. ,I3)
279•             GO TO 1000
280•        805 #RITE(6,810) INUM
281•        810 FORMAT(6X,52H TOO MANY SUBSCRIPTS FOR THIS VARIABLE AT CHAR. NO. ,
282•           $I3)
283•             GO TO 1000
284•        815 #RITE(6,820) INUM
285•        820 FORMAT(6X,51H TOO FE# SUBSCRIPTS FOR THIS VARIABLE AT CHAR. NO. ,
286•           $I3)
287•             GO TO 1000
288•        825 #RITE(6,830) INUM
289•        830 FORMAT(6X,52H ILLEGAL TYPE IN RELATIONAL EXPRESSION AT CHAR. NO. ,
290•           $I3)
291•             GO TO 1000
292•        835 #RITE(6,840) INUM
293•        840 FORMAT(6X,40H TOO MANY ARGUMENTS IN CALLING SEQUENCE ,I3)
294•             GO TO 1000
295•        845 #RITE(6,850)
296•        850 FORMAT(6X,41H TOO MANY FUNCTION REFS IN THIS STATEMENT)
297•             GO TO 1000
298•        855 #RITE(6,860) INUM
299•        860 FORMAT(6X,26H INVALID FORMAL PARAMETER ,A6)
300•             GO TO 1000
301•        865 #RITE(6,870) INUM
302•        870 FORMAT(6X,19H THE FUNCTION NAME ,A6,33H MAY HAVE BEEN PREVIOUSLY M
303•           •ISUSED)
304•             GO TO 1000
305•        875 #RITE(6,880) INUM
306•        880 FORMAT(6X,39H ILLEGAL FIELD DESCRIPTOR AT CHAR. NO. ,I4)
307•             GO TO 1000
308•        885 #RITE(6,890)
309•        890 FORMAT(6X,38H TOO MANY FUNCTION DEFINING STATEMENTS)
310•             GO TO 1000
311•        895 #RITE(6,900)
312•        900 FORMAT(6X,71H TOO MANY EXTERNAL REFERENCES IN THIS STATEMENT - PRO
313•           $CESSING TERMINATED)
314•             GO TO 1000
315•        905 #RITE(6,910)
316•        910 FORMAT(6X,46H STATEMENT IS TOO LONG - PROCESSING TERMINATED)
317•             GO TO 1000
318•        915 #RITE(6,920)
319•        920 FORMAT(6X,46H SESCOMP LIST OVERFLO# - PROCESSING TERMINATED)
320•             GO TO 1000
321•        925 #RITE(6,930)
322•        930 FORMAT(6X,63H OVERFLO# OF INTERFACE DEFINITION TABLE - PROCESSING
323•           $TERMINATED)
324•             GO TO 1000
325•        935 #RITE(6,940)
326•        940 FORMAT(6X,32H TOO MANY EQUIVALENCED VARIABLES)
327•             GO TO 1000
328•        945 #RITE(6,950)
329•        950 FORMAT(6X,61H TOO MANY VARIABLES IN THIS STATEMENT - PROCESSING TE
330•           $RMINATED)
331•       1000 #RITE(6,1)                                                    ERROR153
332•             RETURN                                                       ERROR154
333•             END                                                          ERROR155
```

```
      SUBROUTINE EXPR                                                    EXPR   2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),   RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                   RICH   4
      COMMON/FUNC/IFNCRA(5,17),MARGS,IARGS(50),FNCLOC(5),NFUNC
      COMMON/STRING/NTYPE,NSTR,STR(500)                                  EXPR   5
      COMMON/LIST/NLIST,NINTFC,ISUBLT(3,200),INTFAC(500)
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      INTEGER FNCLOC,OPRA(6),BITPUT,BITGET
      INTEGER D,ASTRIK,DEE,EQUALS,STR,A,COMMA,RPAR,BLANK                 EXPR  10
      DATA (OPRA(I),I=1,6)/1H+,1H-,1H/,1H(,1H),1H,/,ASTRIK/1H*/,DEE/1HD/EXPR  11
     1 ,EQUALS/1H=/,COMMA/1H,/,RPAR/1H)/,LPAR/1H(/,BLANK/1H /           EXPR  12
      LP=0                                                               CH34  29
      NFUNC=0                                                            CH34  30
      K=0
      IEXPST=NBLOCK+1                                                    EXPR  14
      MARGS=0                                                            EXPR  15
  200 K=K+1
      CALL GNLE                                                          EXPR  17
      IF(JTYP .EQ. 0) RETURN                                             EXPR  18
      IF(JTYP .NE. 1) GO TO 20                                           EXPR  19
      IF(LTYP .EQ. 9 .OR. ITYP .EQ. 6) GO TO 2
      IF(ITYP .EQ. 1 .OR. ITYP .EQ. 35) GO TO 1
      GO TO 5
    1 IF(D(1) .EQ. EQUALS) RETURN                                        EXPR  22
      GO TO 5                                                            EXPR  23
    2 IF(D(1) .EQ. RPAR .AND. LP .EQ. 1) RETURN                          EXPR  24
    5 DO 10 I=1,6                                                        EXPR  29
      IF(D(1) .NE. OPRA(I)) GO TO 10                                     EXPR  30
      STR(K)=-I                                                          EXPR  31
      IF(I .EQ. 4) GO TO 6                                               EXPR  32
      IF(I .EQ. 5) GO TO 7                                               EXPR  33
      GO TO 100                                                          EXPR  34
    6 LP=LP+1                                                            EXPR  35
      GO TO 100                                                          EXPR  41
    7 LP=LP-1                                                            EXPR  42
      GO TO 100                                                          EXPR  46
   10 CONTINUE                                                           EXPR  47
      IF(D(1) .NE. EQUALS) GO TO 12                                      EXPR  48
      STR(K)=-18                                                         EXPR  56
      GO TO 100                                                          EXPR  57
   12 IF(D(1) .NE. ASTRIK) GO TO 110                                     EXPR  58
      IF(D(2) .EQ. ASTRIK .AND. M .GT. 1) GO TO 15                       EXPR  59
      STR(K)=-7                                                          EXPR  60
      GO TO 100                                                          EXPR  61
   15 STR(K)=-8                                                          EXPR  62
      GO TO 100                                                          EXPR  63
   20 IF(JTYP .NE. 7) GO TO 30                                           EXPR  64
      IF(LOGID .GT. 9) GO TO 25                                          EXPR  65
      STR(K)=-(LOGID+8)                                                  EXPR  66
      GO TO 100                                                          EXPR  67
   25 STR(K)=LSTART+440000+M*1000000                                     EXPR  68
      GO TO 100                                                          EXPR  69
   30 IF(JTYP .NE. 4) GO TO 40                                           EXPR  70
      IF(IDES .EQ. 0) GO TO 35                                           EXPR  71
      STR(K)=LSTART+420000+M*1000000                                     EXPR  72
      GO TO 100                                                          EXPR  73
   35 CONTINUE                                                           EXPR  74
      STR(K)=LSTART+400000+M*1000000                                     EXPR  75
```

```
60*            GO TO 100                                                EXPR  76
61*         40 IF(JTYP .NE. 6) GO TO 50                                 EXPR  77
62*            STR(K)=LSTART+410000+M*1000000                           EXPR  78
63*            GO TO 100                                                EXPR  79
64*         50 IF(JTYP .NE. 5) GO TO 55
65*            STR(K)=LSTART+430000+M*1000000                           EXPR  81
66*            GO TO 100                                                EXPR  82
67*         55 IF(JTYP .NE. 3) GO TO 60
68*            STR(K)=LSTART+450000+M*1000000
69*            IF(ITYP .NE. 8) CALL ERROR(23,IDM1,IDM2)
70*            GO TO 100
71*         60 IF(JTYP .NE. 2) GO TO 110                                EXPR  83
72*            CALL SEARCH                                              EXPR  84
73*            IBETA=0                                                  EXPR  85
74*            IF(NEXT(JPTR) .NE. LPAR) GO TO 64                        EXPR  86
75*            IF(ISRCH(1) .EQ. 0) GO TO 62                             EXPR  87
76*            IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 1) GO TO 67
77*            CALL SWITCH                                              EXPR  89
78*            IBETA=5                                                  EXPR  90
79*            GO TO 63
80*         62 IBETA=5                                                  EXPR  92
81*            IF(ISRCH(2) .EQ. 1) GO TO 63                             EXPR  93
82*            IDTYP=2                                                  EXPR  94
83*            CALL STORE                                               EXPR  95
84*            LOC=NID                                                  EXPR  96
85*            DO 70 I=1,NLIST
86*            IF(ISUBLT(1,I) .NE. IDTBL(1,LOC)) GO TO 70
87*            IF(BITGET(ISUBLT(2,I),10,4) .NE. 4) GO TO 63
88*            ITP=BITGET(ISUBLT(2,I),13,3)
89*            IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),ITP,10)
90*            IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,11)
91*            GO TO 63                                                 EXPR 101
92*         70 CONTINUE                                                 EXPR 102
93*         63 NFUNC=NFUNC+1                                            EXPR 103
94*            IF(NFUNC .GT. 5) GO TO 120
95*            FNCLOC(NFUNC)=LOC                                        EXPR 104
96*            GO TO 68                                                 EXPR 105
97*         64 IF(ISRCH(2) .NE. 1) GO TO 65                             EXPR 106
98*            IF(NXTID .NE. IFNCNM) CALL ERROR(10,NXTID,KDM2)
99*         65 IF(ISRCH(1) .EQ. 1) GO TO 67
100*           IDTYP=1                                                  EXPR 109
101*           CALL STORE                                               EXPR 110
102*           LOC=NID                                                  EXPR 111
103*           GO TO 68                                                 EXPR 112
104*        67 IBETA=BITGET(IDTBL(3,LOC),7,6)
105*           IF(NXTID .NE. IFNCNM) GO TO 68
106*           IBETA=0
107*           LOC=IDES
108*        68 CALL IMPTYP                                              EXPR 116
109*           IALPH=BITGET(IDTBL(3,LOC),10,3)-1
110*           JPTR=JPTR-1                                              EXPR 118
111*           STR(K)=LOC+10000*IALPH+100000*IBETA+1000000*M            EXPR 119
112*        100 NSTR=K                                                  EXPR 138
113*           IF(NSTR .GT. 500) GO TO 130
114*           GO TO 200
115*        110 CALL ERROR(23,KDM1,KDM2)
116*           RETURN
117*        120 CALL ERROR(90,IDM1,IDM2)
118*           STOP
119*        130 CALL ERROR(91,IDM1,IDM2)
120*           STOP
121*           END                                                      EXPR 141
```

```
      SUBROUTINE EXPRCK                                                 EXPRCK 2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      COMMON/TYP/NQ0,RHSTYP,NQ2,NQ3,LHSTYP
      DIMENSION IA(5,5)                                                 EXPRCK 5
      INTEGER RHSTYP                                                    EXPRCK 6
      DATA ((IA(I,J),I=1,5),J=1,5)/1,0,0,1,0,0,1,0,0,0,1,0,1,1,0,      EXPRCK 7
     1   1,0,1,1,0,0,0,0,0,1/                                          EXPRCK 8
      IF(IA(LHSTYP,RHSTYP+1) .EQ. 0) CALL ERROR(27,KDM1,KDM2)
      RETURN                                                            EXPRCK10
      END                                                               EXPRCK11
```

```
      SUBROUTINE FLOWCK                                                 FLOWCK 2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      COMMON/DOLOOP/ISTACK(4,50),NSTACK,ILOOP,IOVFL#
      COMMON/LABELS/STATRA(2,200),NLABEL                                FLOWCK 6
      COMMON/FLOW/IFL,IRP
      DIMENSION IPATH(100),ISTCK(100)                                   FLOWCK 7
      INTEGER FLWLST(100),BRANCH,STATRA                                 FLOWCK 8
      INTEGER BITPUT,BITGET                                             FLOWCK 9
      EQUIVALENCE (IPATH(1),A(1)),(ISTCK(1),A(101)),(FLWLST(1),A(201))
      IF(IFL .EQ. -1) GO TO 3000
      IRX=0
      NSTCK=0
      NFLOW=0
      NOC=0
      NPTHS=0
      IBLKST=1                                                          FLOWCK11
      CALL CHKLST                                                       CH34  35
      WRITE(6,8)
    8 FORMAT(1H1,38H****** RESULTS OF FLOW ANALYSIS ******//)
    5 DO 10 I=1,NID                                                     FLOWCK13
   10 IDTBL(2,I)=IDTBL(8,I)                                             FLOWCK14
   12 IF(NFLOW .EQ. 0) GO TO 20                                         FLOWCK15
      NOC=0                                                             FLOWCK16
      DO 15 I=1,NFLOW                                                   FLOWCK17
      IF(IABS(FLWLST(I)) .NE. IBLKST) GO TO 15
      NOC=NOC+1                                                         FLOWCK19
   15 CONTINUE                                                          FLOWCK20
      IF(NOC .GT. IRP) GO TO 1500
   20 NFLOW=NFLOW+1                                                     FLOWCK22
      IF(NFLOW .GT. 100) GO TO 4000
      FLWLST(NFLOW)=IBLKST                                              FLOWCK23
      IEND=BITGET(IBLOCK(IBLKST),28,16)-1                               FLOWCK24
      IF(IEND .EQ. -1) IEND=NBLOCK                                      FLOWCK25
      NBR=BITGET(IBLOCK(IBLKST),6,6)                                    FLOWCK26
      ISTART=IEND-NBR+1                                                 FLOWCK27
      IBLKST=NXTBLK(ISTART,IEND)                                        FLOWCK28
      IF(NBR .EQ. 1) GO TO 25                                           FLOWCK29
      FLWLST(NFLOW)=-FLWLST(NFLOW)                                      FLOWCK30
      NSTCK=NSTCK+1                                                     FLOWCK31
      IF(NSTCK .GT. 100) GO TO 5000
      ISTCK(NSTCK)=-NXTBLK(IEND,IEND)                                   FLOWCK32
      IF(NBR .EQ. 2) GO TO 25                                           FLOWCK33
      DO 22 J=3,NBR                                                     FLOWCK34
      NSTCK=NSTCK+1                                                     FLOWCK35
      IF(NSTCK .GT. 100) GO TO 5000
   22 ISTCK(NSTCK)=NXTBLK(IEND-J+2,IEND)                                FLOWCK36
   25 IF(IBLKST .NE. 0) GO TO 12                                        FLOWCK37
      NPATH=0                                                           FLOWCK38
      NPTHS=NPTHS+1                                                     FLOWCK39
      DO 1000 I=1,NFLOW                                                 FLOWCK40
      BRANCH=IABS(FLWLST(I))                                            FLOWCK41
      ISTART=BRANCH+1                                                   FLOWCK42
      NXBLOK=BITGET(IBLOCK(BRANCH),28,16)                               FLOWCK43
      IF(NXBLOK .EQ. 0) NXBLOK=NBLOCK+1                                 FLOWCK44
```

```
      ISL=BITGET(IBLOCK(BRANCH),36,8)                                   FLOWCK45
      ILOOP=BITGET(IBLOCK(BRANCH),12,6)                                 FLOWCK46
      NBR=BITGET(IBLOCK(BRANCH),6,6)                                    FLOWCK47
      IEND=NXBLOK-NBR-1                                                 FLOWCK48
      IF(ISL .EQ. 0) GO TO 45                                           FLOWCK49
      NPATH=NPATH+1                                                     FLOWCK50
      IPATH(NPATH)=STATRA(1,ISL)                                        FLOWCK51
      STATRA(2,ISL)=BITPUT(STATRA(2,ISL),1,18)                          FLOWCK52
   45 IF(IBLOCK(ISTART) .LT. 1000) GO TO 1000                           FLOWCK53
      DO 500 J=ISTART,IEND                                              FLOWCK54
      IT=IBLOCK(J)/1000                                                 FLOWCK55
      LOC=IBLOCK(J)-IT*1000                                             FLOWCK56
      GO TO(50,60,70,80,90,100,200),IT
   50 IF(BITGET(IDTBL(3,LOC),13,1) .EQ. 1) GO TO 120                    FLOWCK58
      IF(IDTBL(2,LOC) .EQ. 2) GO TO 55                                  FLOWCK59
      IF(IDTBL(2,LOC) .EQ. 4) GO TO 180
   52 IDTBL(2,LOC)=1                                                    FLOWCK60
   30 IF(BITGET(IDTBL(3,LOC),17,1) .EQ. 0) GO TO 500
      KTYPE=BITGET(IDTBL(3,LOC),10,3)                                   FLOWCK62
      NXQV=LOC                                                          FLOWCK63
   53 NXQV=IDTBL(7,NXQV)                                                FLOWCK64
      IF(NXQV .EQ. LOC) GO TO 500                                       FLOWCK65
      IF(BITGET(IDTBL(3,NXQV),10,3) .EQ. KTYPE) GO TO 54                FLOWCK66
      IDTBL(2,NXQV)=0                                                   FLOWCK67
      GO TO 53                                                          FLOWCK68
   54 IDTBL(2,NXQV)=1                                                   FLOWCK69
      GO TO 53                                                          FLOWCK70
   55 IF(ILOOP .EQ. 0) GO TO 52                                         FLOWCK71
   57 IF(LOC .EQ. ISTACK(4,ILOOP)) GO TO 110                            FLOWCK72
      IF(ISTACK(3,ILOOP) .EQ. 0) GO TO 52                               FLOWCK73
      ILOOP=ISTACK(3,ILOOP)                                             FLOWCK74
      GO TO 57                                                          FLOWCK75
   60 IF(IDTBL(2,LOC) .EQ. 0) GO TO 140                                 FLOWCK76
      IF(IDTBL(2,LOC) .EQ. 3) GOTO 130                                  FLOWCK77
      GO TO 500                                                         FLOWCK78
   70 IF(BITGET(IDTBL(3,LOC),13,1) .EQ. 1) GO TO 120                    FLOWCK79
      IF(IDTBL(2,LOC) .EQ. 2) GO TO 75                                  FLOWCK80
   72 IDTBL(2,LOC)=2                                                    FLOWCK81
      GO TO 500                                                         FLOWCK82
   75 IF(ILOOP .EQ. 0) GO TO 72                                         FLOWCK83
   77 IF(LOC .EQ. ISTACK(4,ILOOP)) GO TO 110                            FLOWCK84
      IF(ISTACK(3,ILOOP) .EQ. 0) GO TO 72                               FLOWCK85
      ILOOP=ISTACK(3,ILOOP)                                             FLOWCK86
      GO TO 77                                                          FLOWCK87
   80 IF(BITGET(IDTBL(3,LOC),13,1) .EQ. 1) GO TO 120                    FLOWCK88
      IF(IDTBL(2,LOC) .EQ. 2) GO TO 85                                  FLOWCK89
   82 IDTBL(2,LOC)=3                                                    FLOWCK90
      GO TO 500                                                         FLOWCK91
   85 IF(ILOOP .EQ. 0) GO TO 82                                         FLOWCK92
   87 IF(LOC .EQ. ISTACK(4,ILOOP)) GO TO 110                            FLOWCK93
      IF(ISTACK(3,ILOOP) .EQ. 0) GO TO 82                               FLOWCK94
      ILOOP=ISTACK(3,ILOOP)                                             FLOWCK95
      GO TO 87                                                          FLOWCK96
   90 IF(IDTBL(2,LOC) .NE. 3) GO TO 150                                 FLOWCK97
      GO TO 500                                                         FLOWCK98
  100 IDTBL(2,LOC)=0                                                    FLOWCK99
      GO TO 500                                                         FLOWC100
  180 JLOOP=BITGET(IDTBL(3,LOC),36,17)
```

```
      185 IF(ILOOP .EQ. JLOOP) GO TO 160
          ILOOP=ISTACK(3,ILOOP)
          IF(ILOOP .EQ. 0) GO TO 30
          GO TO 185
      200 IF(IDTBL(2,LOC) .EQ. 0) GO TO 140
          IF(IDTBL(2,LOC) .EQ. 3) GO TO 130
          IDTBL(2,LOC)=4
          GO TO 500
      110 IERC=67
          GO TO 400
      120 IERC=68
          GO TO 400
      130 IERC=69
          GO TO 400
      140 IERC=70
          GO TO 400
      150 IERC=71
          GO TO 400
      160 IERC=72
      400 IRX=1
          IF(IDTBL(6,LOC) .EQ. 1) GO TO 500
          IDTBL(6,LOC)=1
          CALL ERROR(IERC,IDTBL(1,LOC),KDM2)
          IF(IERC .NE. 70 .OR. NPATH .EQ. 0) GO TO 500
          WRITE(6,142) (IPATH(K),K=1,NPATH)                              FLOWC116
      142 FORMAT(6X,15H ALONG THE PATH,(10I6))                           FLOWC117
      500 CONTINUE                                                       FLOWC122
     1000 CONTINUE                                                       FLOWC123
     1500 IF(FLWLST(NFLOW) .GT. 0) GO TO 1600                            FLOWC124
          IBLKST=IABS(ISTCK(NSTCK))                                      FLOWC125
          IF(ISTCK(NSTCK) .LT. 0) FLWLST(NFLOW)=-FLWLST(NFLOW)           FLOWC126
          NSTCK=NSTCK-1                                                  FLOWC127
          NOC=0                                                          FLOWC128
          GO TO 5                                                        FLOWC129
     1600 NFLOW=NFLOW-1                                                  FLOWC130
          IF(NFLOW .GT. 0) GO TO 1500                                    FLOWC131
          IF(NLABEL .EQ. 0) GO TO 2010
          DO 2000 J=1,NLABEL                                             FLOWC133
          IF(BITGET(STATRA(2,J),6,6) .EQ. 28) GO TO 2000                 FLOWC134
          IF(BITGET(STATRA(2,J),18,2) .EQ. 1) GO TO 2000
          WRITE(6,1800) STATRA(1,J)
          IRX=1
     1800 FORMAT(6X,57H THERE IS NO COMPLETE PATH THAT CONTAINS STATEMENT NUFLOWC136
         1MBER,I6)                                                       FLOWC137
     2000 CONTINUE                                                       FLOWC138
     2010 IF(IRX .EQ. 0) WRITE(6,2020)
     2020 FORMAT(//6X,16H NO ERRORS FOUND)
          WRITE(6,2100) NPTHS
     2100 FORMAT(//////6X,25H NUMBER OF PATHS CHECKED-,I6)               FLOWC140
          RETURN                                                         FLOWC141
     3000 WRITE(6,3001)
     3001 FORMAT(//31X,57H FLOW ANALYSIS WAS NOT PERFORMED DUE TO ERRORS IN
         *PROGRAM)
          IFL=IRP+1
          RETURN
     4000 WRITE(6,4001)
     4001 FORMAT(//29X,63H TABLE OVERFLOW DURING FLOW ANALYSIS - FLOW ANALYS
         $IS TERMINATED)
          RETURN
     5000 WRITE(6,5001)
     5001 FORMAT(//29X,63H STACK OVERFLOW DURING FLOW ANALYSIS - FLOW ANALYS
         $IS TERMINATED)
          RETURN
          END                                                            FLOWC142
```

```
      SUBROUTINE FNCSTR                                                 FNCSTR 2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,NZ,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      COMMON/FUNC/IFNCRA(5,17),MARGS,IARGS(50),FNCLOC(5),NFUNC
      COMMON/LIST/NLIST,NINTFC,ISUBLT(3,200),INTFAC(500)
      INTEGER FNCLOC,BITPUT,BITGET                                      FNCSTR 6
      IF(NFUNC .EQ. 0) RETURN                                           FNCSTR 7
      DO 40 I=1,NFUNC                                                   FNCSTR 8
      LOC=FNCLOC(I)                                                     FNCSTR 9
      IF(BITGET(IDTBL(3,LOC),20,1) .EQ. 1) GO TO 50
      NARG=IFNCRA(I,1)                                                  FNCSTR10
      IVAR=0
      ITP=BITGET(IDTBL(3,LOC),10,3)
      IF(BITGET(IDTBL(3,LOC),18,1) .EQ. 1) GO TO 20                     FNCSTR11
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,18)                            FNCSTR12
      DO 5 J=1,NLIST
      IF(IDTBL(1,LOC) .NE. ISUBLT(1,J)) GO TO 5
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),J,36)
      LISTLC=J
      GO TO 21
    5 CONTINUE
      CALL ERROR(52,KDM1,KDM2)
      NLIST=NLIST+1
      IF(NLIST .GT. 200) GO TO 60
      ISUBLT(1,NLIST)=IDTBL(1,LOC)
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),NLIST,36)
      IPTR=NINTFC+1
      IF(ITYP .EQ. 8 .AND. I .EQ. 1) ITP=0
      ISUBLT(2,NLIST)=BITPUT(0,ITP,13)
      ISUBLT(2,NLIST)=BITPUT(ISUBLT(2,NLIST),NARG,6)
      ISUBLT(3,NLIST)=IPTR
      NINTFC=IPTR+(NARG-1)/4
      IF(NINTFC .GT. 500) GO TO 70
      DO 10 J=IPTR,NINTFC
   10 INTFAC(J)=IFNCRA(I,J-IPTR+2)
      GO TO 40
   20 LISTLC=BITGET(IDTBL(3,LOC),36,9)
   21 IPTR=ISUBLT(3,LISTLC)
      IF(BITGET(ISUBLT(2,LISTLC),19,1) .EQ. 1) GO TO 22
      NAR2=BITGET(ISUBLT(2,LISTLC),6,6)
      IF(NARG .NE. NAR2) CALL ERROR(26,IDTBL(1,LOC),KDM2)
      NARGS=MIND(NARG,NAR2)
      GO TO 24
   22 IVAR=1
      IF(NARG .LT. 2) CALL ERROR(26,IDTBL(1,LOC),KDM2)
      NARGS=NARG
      ITP1=BITGET(INTFAC(IPTR),3,3)
      NDIM1=BITGET(INTFAC(IPTR),6,3)
   24 IF(ITYP .EQ. 8 .AND. I .EQ. 1) GO TO 25
      IF(BITGET(ISUBLT(2,LISTLC),10,4) .EQ. 4) GO TO 25
      JTP=BITGET(ISUBLT(2,LISTLC),13,3)
      IF(JTP .NE. ITP) CALL ERROR(49,IDTBL(1,LOC),KDM2)
   25 NDPTR=IPTR+(NARGS-1)/4
      KOUNT=0
      DO 32 K=IPTR,NDPTR
      ICOL1=-6
      ICOL2=-3
      DO 32 J=1,4
      KOUNT=KOUNT+1
      IF(KOUNT .GT. NARGS) GO TO 40
      ICOL1=ICOL1+9
      ICOL2=ICOL2+9
      IF(IVAR .EQ. 1) GO TO 26
      ITP1=BITGET(INTFAC(K),ICOL1,3)
      NDIM1=BITGET(INTFAC(K),ICOL2,3)
   26 ITP2=BITGET(IFNCRA(I,K-IPTR+2),ICOL1,3)
      NDIM2=BITGET(IFNCRA(I,K-IPTR+2),ICOL2,3)
      IF(NDIM1 .NE. NDIM2) CALL ERROR(50,KOUNT,KDM2)
```

```
      IF(ITP2 .EQ. 0) GO TO 32
      IF(ITP1 .EQ. 0) GO TO 28
      IF(ITP1 .NE. ITP2) CALL ERROR(51,KOUNT,KDM2)
      GO TO 32
   28 INTFAC(K)=BITPUT(INTFAC(K),ITP2,ICOL1)
   32 CONTINUE
      GO TO 40
   50 CALL STFNC(1)
   40 CONTINUE
      RETURN                                                    FNCSTR48
   60 CALL ERROR(92,IDM1,IDM2)
      STOP
   70 CALL ERROR(93,IDM1,IDM2)
      STOP
      END                                                       FNCSTR49
```

```
      SUBROUTINE FORM
      COMMON/LVARGS/LVFUNC,LVVARG,LVVAD,LVVPOS,LVVTYP,LVVAL,
     +LVHEAD,LVVNVL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP
      COMMON/LVTABL/LVTSIZ,LVMAP(     1)/LVVSEQ/LVSIZE,LVSQSP(     1)
      COMMON /ML/ HOL,ACTION,FUNC1,FUNC2,FUNC3,LEFT,RIGHT,STRING
      COMMON /VAR/ VFOR,NCHAR,NCHARP,CHAR,NDICT
      COMMON /TYP/ NARRAY,TYPE1,TYPE2,ERRFLG
      COMMON /STRING/ NTYPE,NSTR,STR
      INTEGER BITPUT,BITGET
      INTEGER VFOR(15),CHAR,STR(1)
      LOGICAL ERRFLG
C     EXECUTE
      IF(CHAR .NE. IHX) NDICT=-NDICT
      NCHARP=NCHARP+1
      STR(NCHARP)=NDICT
      IF(.NOT. ERRFLG) RETURN
      NCHAR=NCHAR+1
      NC=1+(NCHARP-1)/8
      ICHAR=BITGET(CHAR,6,6)
      VFOR(NC)=BITPUT(VFOR(NC),ICHAR,6*NCHAR)
      IF(NCHAR .EQ. 8) NCHAR=0
      RETURN
      END
```

```
      SUBROUTINE FORMEL                                                 FORM   2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      INTEGER D,BITGET
      DIMENSION ILCHAR(16)
      DATA ILCHAR/1,2,3,4,35,37,38,42,43,44,45,47,58,59,62,63/
      GO TO(100,10,20,100,40,100,100,100),JTYP
   10 CALL CAA(D,M,NXTID)                                               FORM   6
      RETURN                                                            FORM   7
   20 DO 25 I=1,10                                                      FORM   8
      IF(D(I) .NE. IHH) GO TO 25                                        FORM   9
      CALL CAI(D,I-1,N2)                                                FORM  10
      IF(N2 .LT. 1) GO TO 110
      M=N2+1                                                            FORM  11
      IF(M .GT. 500) CALL ERROR(5,KDM1,KDM2)
      JPTR=LSTART+M                                                     FORM  12
      IST=I+1
      DO 22 J=IST,M
      ICHAR=BITGET(D(J),6,6)
      DO 22 K=1,16
      IF(ICHAR .EQ. ILCHAR(K)) GO TO 120
   22 CONTINUE
      IF(ITYP .EQ. 28) RETURN                                           FORM  13
      IF(N2 .GT. 4) CALL ERROR(5,KDM1,KDM2)
      RETURN                                                            FORM  15
   25 CONTINUE                                                          FORM  16
      CALL ERROR(3,KDM1,KDM2)
      RETURN                                                            FORM  18
   40 CALL CAI(D,M,N2)                                                  FORM  21
  100 RETURN                                                            FORM  31
  110 CALL ERROR(7,KDM1,KDM2)
      RETURN
  120 CALL ERROR(23,KDM1,KDM2)
      RETURN
      END                                                               FORM  32
```

```
      SUBROUTINE FRMAT                                                  FRMAT  2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      COMMON/FORMAT/IDESST,IDESND,IGPST,IGPND,IGRP,SEPST,SEPND,
     1 DIR,ICOM,ISEP                                                    FRMAT  5
      DIMENSION RPLOC(20),IALPH(6)                                      FRMAT  6
      INTEGER A,RPLOC,AICH,RPAR,BLANK,SEPST,SEPND,DIR                   FRMAT  7
      DATA BLANK/1H /,AICH/1HH/,LPAR/1H(/,RPAR/1H)/                     FRMAT  8
      DATA (IALPH(I),I=1,6)/1HF,1HO,1HR,1HM,1HA,1HT/                    FRMAT  9
      IFRMT=0                                                           FRMAT 10
      DO 4 I=1,6                                                        FRMAT 11
      IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 70                             FRMAT 12
    4 CONTINUE                                                          FRMAT 13
      NSTART=JPTR                                                       FRMAT 14
      IF(NEXT(JPTR) .NE. LPAR) GO TO 70                                 FRMAT 15
      DO 10 I=1,N                                                       FRMAT 16
      IF(ITYPE(JPTR) .EQ. 2) GO TO 1
      IF(JPTR .GT. N) GO TO 12
      GO TO 10
    1 JPTR=JPTR-1
      CALL GNLE                                                         FRMAT 17
      IF(JTYP .NE. 3) GO TO 10                                          FRMAT 19
      J1=JPTR-1                                                         FRMAT 20
      IH=0                                                              FRMAT 21
      DO 5 J=LSTART,J1                                                  FRMAT 22
      IF(IH .EQ. 1) GO TO 3                                             FRMAT 23
      IF(A(J) .EQ. AICH) IH=1                                           FRMAT 24
      GO TO 5                                                           FRMAT 25
    3 A(J)=BLANK                                                        FRMAT 26
    5 CONTINUE                                                          FRMAT 27
   10 CONTINUE                                                          FRMAT 28
   12 NPAR=0                                                            FRMAT 29
      NRP=0                                                             FRMAT 30
      DO 20 I=NSTART,N                                                  FRMAT 31
      IF(A(I) .NE. LPAR) GO TO 15                                       FRMAT 32
      NPAR=NPAR+1                                                       FRMAT 33
      IF(NPAR .GT. 3) GO TO 70                                          FRMAT 34
      GO TO 20                                                          FRMAT 35
   15 IF(A(I) .NE. RPAR) GO TO 20                                       FRMAT 36
      NPAR=NPAR-1                                                       FRMAT 37
      NRP=NRP+1                                                         FRMAT 38
      RPLOC(NRP)=I                                                      FRMAT 39
      IF(NPAR .LT. 0) GO TO 70                                          FRMAT 40
   20 CONTINUE                                                          FRMAT 41
      IF(NPAR .NE. 0) GO TO 70                                          FRMAT 42
      IF(NEXT(RPLOC(NRP)+1) .NE. BLANK) GO TO 70                        FRMAT 43
      DO 60 I=1,NRP                                                     FRMAT 44
      IGPND=RPLOC(I)                                                    FRMAT 45
      DO 25 J=1,N                                                       FRMAT 46
      K=IGPND-J                                                         FRMAT 47
      IF(A(K) .NE. LPAR) GO TO 25                                       FRMAT 48
      IGPST=K                                                           FRMAT 49
      GO TO 30                                                          FRMAT 50
   25 CONTINUE                                                          FRMAT 51
   30 CALL GROUP                                                        FRMAT 52
      IF(IGRP .EQ. 0) RETURN                                            FRMAT 53
      IF(I .EQ. NRP) GO TO 65                                           FRMAT 54
      JPTR=IGPST-1                                                      FRMAT 55
   31 CONTINUE
      IF(IPREV(JPTR) .EQ. 2) GO TO 31
      IGPST=JPTR+2                                                      FRMAT 57
      GO TO 35                                                          FRMAT 58
   35 SEPST=IGPND+1                                                     FRMAT 60
      DIR=1
      CALL SEPAR                                                        FRMAT 62
      IF(ISEP .NE. 1) GO TO 40                                          FRMAT 63
      IGPND=SEPND                                                       FRMAT 64
```

```
      GO TO 50                                                          FRMAT 65
   40 IF(NEXT(SEPST) .NE. RPAR) GO TO 70                                FRMAT 66
      SEPST=IGPST-1                                                     FRMAT 67
      DIR=-1
      CALL SEPAR                                                        FRMAT 69
      IF(ISEP .NE. 1) GO TO 45                                          FRMAT 70
      IGPST=SEPND                                                       FRMAT 71
      GO TO 50                                                          FRMAT 72
   45 IF(A(SEPND) .NE. LPAR) GO TO 70                                   FRMAT 73
   50 DO 55 J=IGPST,IGPND                                               FRMAT 74
      A(J)=BLANK                                                        FRMAT 75
   55 CONTINUE                                                          FRMAT 76
   60 CONTINUE                                                          FRMAT 77
   65 IFRMT=1                                                           FRMAT 78
      RETURN                                                            FRMAT 79
   70 CALL ERROR(7,KDM1,KDM2)
      RETURN                                                            FRMAT 81
      END                                                               FRMAT 82
```

```
      SUBROUTINE GENROL                                                 GENROL  2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH    2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH    4
      COMMON/GLOBAL/NBLK,NREF,NSUBS,BLKTBL(200),EXTTBL(100),ISUBS(100)
      COMMON/LIST/NLIST,NINTFC,ISUBLT(3,200),INTFAC(500)
      COMMON/INPOUT/NCALL,IN,IOP                                        GENROL  6
      COMMON/WASTE/IDUM(63)
      INTEGER BLKTBL,EXTTBL,BITGET
      WRITE(6,2)                                                        GENROL  9
    2 FORMAT(1H1)                                                       GENROL10
C     IF CDC 6700 - PLACE A :C: IN COLUMN 1 OF NEXT CARD
      WRITE(IOP,50)
   50 FORMAT(19H@FOR,IS ROLCAL)
C     IF UNIVAC 1108 - PLACE A 'C' IN COLUMN ONE OF NEXT CARD
C     WRITE(IOP,4)
    4 FORMAT(5X,48H PROGRAM ROLCAL(OUTPUT,TAPE6=OUTPUT,TAPE3,TAPE9))    GENROL12
      IF(NBLK .EQ. 0) GO TO 6                                           GENROL13
      K=-1                                                              GENROL14
      DO 5 I=1,NBLK                                                     GENROL15
      K=K+1                                                             GENROL17
      INDEX=BLKTBL(I)
      ISZ=BITGET(ISUBLT(2,INDEX),36,15)
      WRITE(IOP,3) ISUBLT(1,INDEX),K,ISZ
    3 FORMAT(5X,8H COMMON/,A6,3H/IX,I2,1H(,I6,1H))                      GENROL19
    5 CONTINUE                                                          GENROL20
    6 WRITE(IOP,7) MODE
    7 FORMAT(5X,4H J=1/5X,6H MODE=,I1/5X,13H DO 10 I=1,13/5X,6H J=J-1)
      IF(NBLK .EQ. 0) GO TO 22                                          GENROL27
      K=-1                                                              GENROL28
      DO 20 I=1,NBLK                                                    GENROL29
      K=K+1                                                             GENROL31
      KK=1000*K                                                         GENROL32
      INDEX=BLKTBL(I)
      ISZ=BITGET(ISUBLT(2,INDEX),36,15)
      WRITE(IOP,10) KK,ISZ,K,KK
   10 FORMAT(5X,4H DO ,I4,5H K=1,,I6/5X,3H IX,I2,5H(K)=1/1X,I4,
     $ 9H CONTINUE)
      IF(ISUBLT(1,INDEX) .NE. 6HSESCOM) GO TO 20
      WRITE(IOP,15) K,K,K
   15 FORMAT(5X,3H IX,I2,7H(17)=10/5X,3H IX,I2,7H(20)=11/
     * 5X,3H IX,I2,7H(23)=12)
   20 CONTINUE                                                          GENROL39
   22 NARG=BITGET(IDTBL(3,I),7,6)
      DO 30 I=1,NARG                                                    GENROL41
      IF(I .EQ. NARG) GO TO 25                                          GENROL42
      IDUM(I)=2HD,                                                      GENROL43
      GO TO 30                                                          GENROL44
   25 IDUM(I)=2HJ)                                                      GENROL45
   30 CONTINUE                                                          GENROL46
      WRITE(IOP,35) IDTBL(1,I),(IDUM(I),I=1,NARG)
   35 FORMAT(5X,6H CALL ,A6,1H(,41A2/5X,1H1,1X,22A2)                    GENROL48
      WRITE(IOP,38)
   38 FORMAT(5X,24H IF(MODE .EQ. 3) GO TO 5)
      WRITE(IOP,40)
   40 FORMAT(5X,14H CALL MODID(J)/3X,12H 5 ENDFILE 3/2X,12H 10 CONTINUE/
     * 5X,12H CALL CMPARE/5X,11H ENDFILE 13/5X,11H ENDFILE 14/
     * 5X,11H ENDFILE 15/5X,10H REWIND 13/5X,10H REWIND 14/
     * 5X,10H REWIND 15/5X,5H STOP/5X,4H END)
      RETURN                                                            GENROL59
      END                                                               GENROL60
```

```
      SUBROUTINE GLOTAB                                                 GLOTAB 2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/GLOBAL/NBLK,NREF,NSUBS,BLKTBL(200),EXTTBL(100),ISUBS(100)
      COMMON/LIST/NLIST,NINTFC,ISUBLT(3,200),INTFAC(500)
      INTEGER BITGET,BLKTBL,EXTTBL,KLAS(4,7)
      DATA KLAS/6HUSER S,6HUPPLIE,1HD,1H ,6HSUBROU,6HTINE M,5HODULE,1H ,
     $ 6HFUNCTI,6HON MOD,3HULE,1H ,6HANCILL,6HARY SU,6HBPROGR,2HAM,
     $ 6HANSI F,6HUNCTIO,1HN,1H ,6HMAIN P,6HROGRAM,1H ,1H ,6HEXTRAO,
     $ 6HRDINAR,6HY SUBP,6HROGRAM/
      WRITE(6,1)                                                        GLOTAB 8
    1 FORMAT(1H1,48X,23H GLOBAL REFERENCE TABLE)                        GLOTAB 9
      IF(NREF .EQ. 0) GO TO 25                                          GLOTAB10
      WRITE(6,2)                                                        GLOTAB11
    2 FORMAT(//50X,20H EXTERNAL REFERENCES)                             GLOTAB12
      DO 20 I=1,NREF                                                    GLOTAB13
      INDEX=EXTTBL(I)
      J=BITGET(ISUBLT(2,INDEX),10,4)
      WRITE(6,10) ISUBLT(1,INDEX),(KLAS(K,J+1),K=1,4)
   10 FORMAT(45X,A6,4X,4A6)
   20 CONTINUE
   25 IF(NBLK .EQ. 0) GO TO 40
      INDEX=BLKTBL(1)
      IF(NBLK .EQ. 1 .AND. ISUBLT(1,INDEX) .EQ. 1H ) GO TO 40
      WRITE(6,30)
   30 FORMAT(//49X,23H LABELLED COMMON BLOCKS/43X,11H BLOCK NAME,7X,
     $ 5H SIZE,7X,6H CLASS)
      DO 38 J=1,NBLK
      INDEX=BLKTBL(J)
      ICAT=BITGET(ISUBLT(2,INDEX),10,4)-6
      ISZ=BITGET(ISUBLT(2,INDEX),36,15)
   35 FORMAT(46X,A6,5X,I8,5X,9HCATEGORY ,I2)
      WRITE(6,35) ISUBLT(1,INDEX),ISZ,ICAT
   38 CONTINUE
   40 WRITE(6,45)                                                       GLOTAB34
   45 FORMAT(///48X,24H SUBROUTINES ENCOUNTERED)                        GLOTAB35
      DO 60 I=1,NSUBS
C** GET SUBROUTINE CLASS
      INDEX=ISUBS(I)
      J=BITGET(ISUBLT(2,INDEX),10,4)
      WRITE(6,10) ISUBLT(1,INDEX),(KLAS(K,J+1),K=1,4)
   60 CONTINUE
      RETURN                                                            GLOTAB38
      END                                                               GLOTAB39
```

```
      SUBROUTINE GNLE                                                   GNLE   2
      COMMON A(1326),D(500),,DTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     • JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      COMMON/LOGIC/LOG,LOGST                                            GNLE   4
      COMMON/REALNO/IREAL,IRELND,IP                                     GNLE   5
      INTEGER A,D,BLANK,PLUS,EQUALS,SLASH,RPAR,COMMA,ASTRIK,            GNLE   6
     1 AICH,DECPT                                                       GNLE   7
      DATA BLANK/1H /,PLUS/1H+/,MINUS/1H-/,EQUALS/1H=/,SLASH/1H//,      GNLE   8
     1 RPAR/1H)/,COMMA/1H,/,ASTRIK/1H*/,AICH/1HH/,LPAR/1H(/,DECPT/1H./  GNLE   9
      JTYP=0                                                            GNLE  10
      NXT=NEXT(JPTR)                                                    GNLE  11
      IF(NXT .EQ. BLANK) RETURN                                         GNLE  12
      LSTART=JPTR-1                                                     GNLE  13
      IF(NXT .EQ. PLUS) GO TO 1                                         GNLE  14
      IF(NXT .EQ. RPAR) GO TO 1                                         GNLE  15
      IF(NXT .EQ. MINUS) GO TO 1                                        GNLE  16
      IF(NXT .EQ. SLASH) GO TO 1                                        GNLE  17
      IF(NXT .EQ. COMMA) GO TO 1                                        GNLE  18
      IF(NXT .EQ. EQUALS) GO TO 1                                       GNLE  19
      GO TO 2                                                           GNLE  20
    1 JTYP=1                                                            GNLE  21
      M=1                                                               GNLE  22
      GO TO 90                                                          GNLE  23
    2 IF(NXT .NE. ASTRIK) GO TO 4                                       GNLE  24
      IF(NEXT(JPTR) .NE. ASTRIK) GO TO 1                                GNLE  25
      M=2                                                               GNLE  26
      JTYP=1                                                            GNLE  27
      GO TO 90                                                          GNLE  28
    4 IF(NXT .NE. LPAR) GO TO 40                                        GNLE  29
      IF(LSTART .EQ. 1) GO TO 10                                        GNLE  30
      IM1=LSTART-1                                                      GNLE  31
      IF(IPREV(IM1) .NE. 3) GO TO 1                                     GNLE  32
   10 CONTINUE                                                          GNLE  33
      NXT=NEXT(LSTART+1)                                                GNLE  34
      IF(NXT .EQ. BLANK) GO TO 120                                      GNLE  35
      IF(NXT .NE. PLUS .AND. NXT .NE. MINUS) GO TO 22                   GNLE  36
      IP=JPTR                                                           GNLE  37
      GO TO 24                                                          GNLE  38
   22 IP=JPTR-1                                                         GNLE  39
   24 CALL REALCK                                                       GNLE  40
      IF(IREAL .EQ. 0) GO TO 1                                          GNLE  41
      IF(IDES .EQ. 1) GO TO 1                                           GNLE  42
      IF(NEXT(IRELND+1) .NE. COMMA) GO TO 1                             GNLE  43
      NXT=NEXT(JPTR)                                                    GNLE  44
      IF( NXT .NE. PLUS .AND. NXT .NE. MINUS) GO TO 30                  GNLE  45
      IP=JPTR                                                           GNLE  46
      GO TO 35                                                          GNLE  47
   30 IP=JPTR-1                                                         GNLE  48
   35 CALL REALCK                                                       GNLE  49
      IF(IREAL .EQ. 0) GO TO 120                                        GNLE  50
      IF(IDES .EQ. 1) GO TO 120                                         GNLE  51
      IF(NEXT(IRELND+1) .NE. RPAR) GO TO 120                            GNLE  52
      JTYP=6                                                            GNLE  53
      M=JPTR-LSTART                                                     GNLE  54
      GO TO 90                                                          GNLE  55
   40 IF(NXT .NE. DECPT) GO TO 50                                       GNLE  56
      ITP=ITYPE(JPTR)                                                   GNLE  57
      GO TO (42,44,120),ITP                                             GNLE  58
   42 LOGST=LSTART+1                                                    GNLE  59
      CALL LOGCHK                                                       GNLE  60
      IF(LOG .EQ. 0) GO TO 120                                          GNLE  61
      JTYP=7                                                            GNLE  62
      M=JPTR-LSTART                                                     GNLE  63
      GO TO 90                                                          GNLE  64
   44 IP=LSTART                                                         GNLE  65
```

```
      CALL REALCK                                                    GNLE  66
      IF(IREAL .EQ. 0) GO TO 120                                     GNLE  67
      JTYP=4                                                         GNLE  68
      M=IRELND-LSTART+1                                              GNLE  69
      GO TO 90                                                       GNLE  70
   50 CONTINUE                                                       GNLE  71
      IF(ITYPE(LSTART) .NE. 2) GO TO 85                              GNLE  72
      IF(ITYP .EQ. 28) GO TO 54                                      GNLE  73
      IP=LSTART                                                      GNLE  74
      CALL REALCK                                                    GNLE  75
      IF(IREAL .EQ. 0) GO TO 54                                      GNLE  76
      JTYP=4                                                         GNLE  77
      M=IRELND-LSTART+1                                              GNLE  78
      GO TO 90                                                       GNLE  79
   54 JPTR=LSTART+1                                                  GNLE  80
   55 IF(ITYPE(JPTR) .EQ. 2) GO TO 57                                GNLE  81
      GO TO 65                                                       GNLE  82
   57 IF(JPTR .GT. N) GO TO 60                                       GNLE  83
      GO TO 55                                                       GNLE  84
   60 M=N-LSTART+1                                                   GNLE  85
      JTYP=5                                                         GNLE  86
      GO TO 90                                                       GNLE  87
   65 IF(A(JPTR-1) .NE. AICH) GO TO 67                               GNLE  88
      IF(ITYP .EQ. 8 .OR. ITYP .EQ. 28 .OR. ITYP .EQ. 27) GO TO 70   GNLE  89
   67 M=JPTR-LSTART-1                                                GNLE  90
      JTYP=5                                                         GNLE  91
      GO TO 90                                                       GNLE  92
   70 IF(JPTR .GT. N) GO TO 120                                      GNLE  93
      M=N-LSTART+1                                                   GNLE  94
      IF(M .GT. 500) M=500
      JTYP=3                                                         GNLE  95
      GO TO 90                                                       GNLE  96
   85 CONTINUE                                                       GNLE  97
      IF(ITYPE(LSTART) .NE. 1) GO TO 120                             GNLE  98
      IF(ITYP .EQ. 28) GO TO 100
   88 CONTINUE                                                       GNLE  99
      IF(ITYPE(JPTR) .NE. 3) GO TO 86                                GNLE 100
      M=JPTR-LSTART-1                                                GNLE 101
      JTYP=2                                                         GNLE 102
      GO TO 90                                                       GNLE 103
   86 IF(JPTR .GT. N) GO TO 87                                       GNLE 104
      GO TO 88                                                       GNLE 105
  100 M=1
      JTYP=2
      GO TO 90
   87 M=N-LSTART+1                                                   GNLE 106
      JTYP=2                                                         GNLE 107
   90 CONTINUE                                                       GNLE 108
      DO 91 L=1,M                                                    GNLE 109
      LL=LSTART+L-1                                                  GNLE 110
      D(L)=A(LL)                                                     GNLE 111
   91 CONTINUE                                                       GNLE 112
      JPTR=LSTART+M                                                  GNLE 113
      CALL SQUEEZ
      CALL FORMEL                                                    GNLE 115
      RETURN                                                         GNLE 116
  120 CONTINUE                                                       GNLE 117
      JTYP=8                                                         GNLE 118
      RETURN                                                         GNLE 119
      END                                                            GNLE 120
```

```
      SUBROUTINE GOTO                                                   GOTO   2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      COMMON/LABELS/STATRA(2,200),NLABEL                                GOTO   4
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      DIMENSION IALPH(4)                                                GOTO   6
      INTEGER STATRA,BLANK                                              GOTO   7
      INTEGER BITPUT                                                    GOTO   8
      DATA (IALPH(I),I=1,4)/1HG,1HO,1HT,1HO/                            GOTO   9
      DATA BLANK/1H /                                                   GOTO  10
      DO 5 I=1,4                                                        GOTO  11
      IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 10                             GOTO  12
    5 CONTINUE                                                          GOTO  13
      CALL GNLE                                                         GOTO  14
      IF(JTYP .NE. 5) GO TO 10                                          GOTO  15
      CALL STSRCH                                                       GOTO  16
      STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,12)                          GOTO  17
      IF(NEXT(JPTR) .NE. BLANK) GO TO 10                                GOTO  18
      NBLOCK=NBLOCK+1                                                   GOTO  19
      IBLOCK(NBLOCK)=LOC                                                GOTO  20
      NBRNCH=1                                                          GOTO  21
      NB=1                                                              GOTO  22
      RETURN                                                            GOTO  23
   10 CALL ERROR(7,KDM1,KDM2)
      RETURN                                                            GOTO  25
      END                                                               GOTO  26
```

```
      SUBROUTINE GROUP                                                  GROUP  2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      COMMON/FORMAT/IDESST,IDESND,IGPST,IGPND,IGRP,SEPST,SEPND,
     1 DIR,ICOM,ISEP                                                    GROUP  5
      INTEGER A,RPAR,SEPST,SEPND,DIR                                    GROUP  6
      DATA RPAR/1H)/                                                    GROUP  7
      IF(NEXT(IGPST+1) .EQ. RPAR) GO TO 20                              GROUP  8
      SEPST=JPTR-1                                                      GROUP  9
      DIR=1
      CALL SEPAR                                                        GROUP 11
      IF(ISEP .EQ. -1 .OR. ICOM .EQ. 1) GO TO 30                        GROUP 12
      IF(ISEP .EQ. 0) IDESST=SEPST                                      GROUP 13
      IF(ISEP .EQ. 1) IDESST=SEPND+1                                    GROUP 14
      IF(NEXT(IDESST) .EQ. RPAR) GO TO 20                               GROUP 15
      SEPST=IGPND-1                                                     GROUP 16
      DIR=-1
      CALL SEPAR                                                        GROUP 18
      IF(ISEP .EQ. -1 .OR. ICOM .EQ. 1) GO TO 30                        GROUP 19
      DIR=1
   10 CONTINUE                                                          GROUP 21
      CALL DESCRP                                                       GROUP 22
      IF(IDES .EQ. 0) GO TO 40
      SEPST=IDESND+1                                                    GROUP 24
      IF(NEXT(SEPST) .EQ. RPAR) GO TO 20                                GROUP 25
      CALL SEPAR                                                        GROUP 26
      IF(ISEP .EQ. 0 .OR. ISEP .EQ. -1) GO TO 30                        GROUP 27
      IDESST=SEPND+1                                                    GROUP 28
      IF(NEXT(IDESST) .NE. RPAR) GO TO 10                               GROUP 29
   20 IGRP=1                                                            GROUP 30
      RETURN                                                            GROUP 31
   30 IGRP=0                                                            GROUP 32
      CALL ERROR(7,KDM1,KDM2)
      RETURN                                                            GROUP 34
   40 CALL ERROR(88,IDESST,KDM2)
      IGRP=0
      RETURN
      END                                                               GROUP 35
```

```
      SUBROUTINE GRT                                                    GRT    2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      COMMON/GLOBAL/NBLK,NREF,NSUBS,BLKTBL(200),EXTTBL(100),ISUBS(100)
      COMMON/LIST/NLIST,NINTFC,ISUBLT(3,200),INTFAC(500)
      INTEGER EXTTBL,BLKTBL,BLANK,BITPUT,BITGET
      DATA BLANK/1H /
      WRITE(6,1)
    1 FORMAT(//)
      ISUB=INITID(2)                                                    GRT   10
      IF(ISUB .EQ. 0) GO TO 15                                          GRT   11
   10 ISUB=IDTBL(2,ISUB)
      IF(ISUB .EQ. 0) GO TO 15
      IF(BITGET(IDTBL(3,ISUB),20,1) .EQ. 1) GO TO 10
      IF(NREF .EQ. 0) GO TO 4                                           GRT   16
      DO 3 K=1,NREF                                                     GRT   17
      INDEX=EXTTBL(K)
      IF(IDTBL(1,ISUB) .EQ. ISUBLT(1,INDEX)) GO TO 10
    3 CONTINUE                                                          GRT   41
    4 NREF=NREF+1                                                       GRT   42
      IF(NREF .GT. 100) GO TO 50
      EXTTBL(NREF)=BITGET(IDTBL(3,ISUB),36,9)
      IF(MODE .EQ. 1) GO TO 10
      INDEX=EXTTBL(NREF)
      KLAS=BITGET(ISUBLT(2,INDEX),10,4)
      IF(KLAS .EQ. 1 .OR. KLAS .EQ. 2) WRITE(9) IDTBL(1,ISUB)
      GO TO 10
   15 IBLK=INITID(3)
   20 IF(IBLK .EQ. 0) RETURN
      IF(IDTBL(1,IBLK) .EQ. 1H ) GO TO 45
      DO 25 I=1,NLIST
      IF(IDTBL(1,IBLK) .NE. ISUBLT(1,I)) GO TO 25
      LISTLC=I
      IF(BITGET(ISUBLT(2,I),10,4) .EQ. 7) GO TO 30
      IF(BITGET(ISUBLT(2,I),36,15) .NE. 0) GO TO 30
      ISZ=IDTBL(4,IBLK)
      ISUBLT(2,I)=BITPUT(ISUBLT(2,I),ISZ,36)
      GO TO 30
   25 CONTINUE
      CALL ERROR(62,IDTBL(1,IBLK),KDM2)
      NLIST=NLIST+1
      ISUBLT(1,NLIST)=IDTBL(1,IBLK)
      ISZ=IDTBL(4,IBLK)
      ISUBLT(2,NLIST)=BITPUT(ISZ,10,10)
      LISTLC=NLIST
   30 IDTBL(3,IBLK)=BITPUT(IDTBL(3,IBLK),LISTLC,36)
      IF(NBLK .EQ. 0) GO TO 40
      DO 35 K=1,NBLK
      IF(LISTLC .EQ. BLKTBL(K)) GO TO 45
   35 CONTINUE
   40 NBLK=NBLK+1
      IF(NBLK .GT. 200) GO TO 60
      BLKTBL(NBLK)=LISTLC
   45 IBLK=IDTBL(2,IBLK)
      GO TO 20
   50 CALL ERROR(59,KDM1,KDM2)
      STOP
   60 CALL ERROR(60,KDM1,KDM2)
      STOP
      END                                                               GRT   65
```

```
      SUBROUTINE IMPTYP                                                 IMPTYP 2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      DIMENSION IALPH(6)                                                IMPTYP 4
      INTEGER D,BITPUT,BITGET                                           IMPTYP 5
      DATA (IALPH(I),I=1,6)/IHI,IHJ,IHK,IHL,IHM,IHN/                    IMPTYP 6
      IF(BITGET(IDTBL(3,LOC),11,1) .EQ. 1) GO TO 20
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,11)                            IMPTYP 8
      DO 10 I=1,6                                                       IMPTYP 9
      IF(D(I) .NE. IALPH(I)) GO TO 10                                   IMPTYP10
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),4,10)
      GO TO 20
   10 CONTINUE                                                          IMPTYP13
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,10)
   20 IF(ITYP .LE. 18) IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,19)
      RETURN                                                            IMPTYP17
      END                                                               IMPTYP18
```

```
      SUBROUTINE INIT                                                    INIT   2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),   RICH   2
     • JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                   RICH   4
      COMMON/FUNC/IFNCRA(5,17),MARGS,IARGS(50),FNCLOC(5),NFUNC
      COMMON/STRING/NTYPE,NSTR,STR(500)                                  INIT   5
      COMMON/TYP/NQQ,RHSTYP,NQ2,NQ3,LHSTYP
      COMMON/LIST/NLIST,NINTFC,ISUBLT(3,208),INTFAC(500)
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      COMMON/STFUNC/NSTFNC,ISTFNC(10)
      INTEGER RHSTYP
      INTEGER A,EQUALS,COMMA,RPAR,STR                                    INIT   10
      INTEGER BITPUT,BITGET,FNCLOC                                       INIT   11
      DATA EQUALS/1H=/,LPAR/1H(/,COMMA/1H,/,RPAR/1H)/                    INIT   12
      IFN=0                                                              INIT   13
      NTYPE=1                                                            INIT   14
      IPTR=JPTR                                                          INIT   15
      CALL GNLE                                                          INIT   16
      IF(JTYP .NE. 2) GO TO 40                                           INIT   17
      CALL SEARCH                                                        INIT   18
      IF(NEXT(JPTR) .NE. EQUALS) GO TO 6                                 INIT   19
      LOC2=0                                                             INIT   20
      IF(ISRCH(2) .NE. 1) GO TO 2                                        INIT   21
      IF(NXTID .NE. IFNCNM) CALL ERROR(10,NXTID,KDM2)
      IFN=1                                                              INIT   23
      CALL IMPTYP                                                        INIT   24
      LOC2=LOC                                                           INIT   25
    2 IF(ISRCH(1) .NE. 1) GO TO 18
      IF(LOC2 .EQ. 0) GO TO 4
      LOC=IDES
      GO TO 5
   18 IDTYP=1
      CALL STORE                                                         INIT   28
      LOC=NID                                                            INIT   29
      IF(LOC2 .EQ. 0) GO TO 4
      IDTBL(3,LOC)=IDTBL(3,LOC2)                                         INIT   31
      GO TO 5                                                            INIT   32
    4 CALL IMPTYP                                                        INIT   33
    5 LHSTYP=BITGET(IDTBL(3,LOC),10,3)                                   INIT   34
      IF(LHSTYP .EQ. 5) NTYPE=2
      GO TO 30                                                           INIT   36
    6 IF(A(JPTR-1) .NE. LPAR) GO TO 40                                   INIT   37
      IF(ISRCH(1) .EQ. 1) GO TO 12                                       INIT   38
      IF(ISRCH(2) .NE. 1) GO TO 15                                       INIT   39
      CALL ERROR(10,NXTID,KDM2)
      GO TO 8                                                            INIT   41
   15 IDTYP=2                                                            INIT   42
      CALL STORE                                                         INIT   43
      LOC=NID                                                            INIT   44
      GO TO 8                                                            INIT   45
    7 CALL SWITCH                                                        INIT   46
```

```
52.          8 CALL IMPTYP                                                  INIT  47
53.            LHSTYP=BITGET(IDTBL(3,LOC),10,3)                             INIT  48
54.            IF(LHSTYP .EQ. 5) NTYPE=2
55.            NARG=0                                                       INIT  50
56.            ITYP=35                                                      INIT  51
57.            NSTFNC=NSTFNC+1
58.            IF(NSTFNC .GT. 10) GO TO 60
59.            ISTFNC(NSTFNC)=LOC
60.            LOC1=LOC
61.            IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,20)
62.         10 CALL GNLE                                                    INIT  52
63.            IF(JTYP .NE. 2) GO TO 50                                     INIT  53
64.            NARG=NARG+1                                                  INIT  54
65.            CALL SEARCH
66.            IF(ISRCH(2) .EQ. 1) CALL ERROR(54,NARG,KDM2)
67.            IF(ISRCH(1) .EQ. 1) GO TO 20
68.            IDTYP=1
69.            CALL STORE
70.            LOC=NID
71.            GO TO 25
72.         20 IF(BITGET(IDTBL(3,LOC),1,1) .EQ.    CALL ERROR( ),NARG,KDM2)
73.         25 CALL IMPTYP
74.            IF(NEXT(JPTR) .EQ. COMMA) GO TO 10                           INIT  55
75.            IF(A(JPTR-1) .NE. RPAR) GO TO 40                             INIT  56
76.            IDTBL(3,LOC1)=BITPUT(IDTBL(3,LOC1),NARG,7)
77.            IF(NEXT(JPTR) .EQ. EQUALS) GO TO 32                          INIT  58
78.            GO TO 40                                                     INIT  59
79.         12 IF(BITGET(IDTBL(3,LOC),1,1) .NE. 1) GO TO 7                  INIT  60
80.            CALL IMPTYP                                                  INIT  61
81.            LHSTYP=BITGET(IDTBL(3,LOC),10,3)                             INIT  62
82.            IF(LHSTYP .EQ. 5) NTYPE=2
83.            JPTR=IPTR                                                    INIT  64
84.            JBLOCK=NBLOCK+1                                              INIT  65
85.            CALL EXPR                                                    INIT  66
86.            CALL PARSE                                                   INIT  67
87.            CALL BLKSTR
88.            IBLOCK(JBLOCK)=IBLOCK(JBLOCK) - 1000                         INIT  68
89.            GO TO 32                                                     INIT  69
90.         30 NBLOCK=NBLOCK+1                                              INIT  70
91.            JBLOCK=NBLOCK                                                INIT  71
92.            IBLOCK(NBLOCK)=1000+LOC                                      INIT  72
93.         32 NTMS=0                                                       INIT  73
94.            IPTR=JPTR
95.            CALL EXPR                                                    INIT  74
96.            IF(JPTR .LE. N) CALL ERROR(7,KDM1,KDM2)
97.            CALL PARSE                                                   INIT  75
98.            CALL FNCSTR                                                  INIT  76
99.            CALL EXPRCK                                                  INIT  77
100.           IF(ITYP .EQ. 35) GO TO 36                                    INIT  78
101.           CALL BLKSTR
102.           IBLOCK(NBLOCK+1)=IBLOCK(JBLOCK)                              INIT  79
103.           DO 34 K=JBLOCK,NBLOCK                                        INIT  80
104.        34 IBLOCK(K)=IBLOCK(K+1)                                        INIT  81
105.           IF(LTYP .EQ. 1) RETURN
106.        36 IF(MODE .NE. 1) GO TO 35
107.           IF(RHSTYP .EQ. 3 .OR. RHSTYP .EQ. 4) RETURN
108.           IF(NSTR .LT. 6) RETURN
109.           JPTR=IPTR-1
```

```
110•          CALL CNVRT                                                    INIT  84
111•          RETURN                                                        INIT  85
112•       35 IF(NFUNC .EQ. 0) RETURN                                       INIT  86
113•          IF(IFN .EQ. 1) RETURN                                         INIT  87
114•          DO 38 J=1,NFUNC                                               INIT  88
115•          LOC=FNCLOC(J)
116•          INDEX=BITGET(IDTBL(3,LOC),36,9)
117•          KLAS=BITGET(ISUBLT(2,INDEX),10,4)
118•          IF(KLAS .NE. 1 .AND. KLAS .NE. 2) GO TO 38
119•          CALL CALL2                                                    INIT  92
120•          RETURN                                                        INIT  93
121•       38 CONTINUE                                                      INIT  95
122•          RETURN                                                        INIT  96
123•       40 CALL ERROR(7,KDM1,KDM2)
124•          RETURN                                                        INIT  98
125•       50 CALL ERROR(15,KDM1,KDM2)
126•          RETURN                                                        INIT 100
127•       60 CALL ERROR(89,IDM1,IDM2)
128•          RETURN
129•          END                                                           INIT 101
```

```
  1•          SUBROUTINE INTRIN
  2•          COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),
  3•         • JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
  4•         2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES
  5•          COMMON/LIST/NLIST,NINTFC,ISUBLT(3,200),INTFAC(500)
  6•          INTEGER BITGET
  7•          DO 100 I=1,NLIST
  8•          IF(BITGET(ISUBLT(2,I),10,4) .NE. 4) GO TO 100
  9•          NXTID=ISUBLT(1,I)
 10•          CALL SEARCH
 11•          CALL COMSCH
 12•          IF(ISRCH(1) .EQ. 1 .OR. ISRCH(3) .EQ. 1) CALL ERROR(74,NXTID,KDM2)
 13•       100 CONTINUE
 14•          RETURN
 15•          END
```

```
      SUBROUTINE IO                                                    IO     2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                 RICH   4
      COMMON/STRING/NTYPE,NSTR,STR(500)                                IO     4
      COMMON/LABELS/STATRA(2,200),NLABEL                               IO     5
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      COMMON/INPOUT/NCALL,IN,IOP                                       IO     7
      DIMENSION IALPH1(4),IALPH2(5),IALPH3(8)                          IO     8
      INTEGER A,STATRA,RPAR,COMMA,BLANK                                IO     9
      INTEGER BITPUT,BITGET                                            IO    10
      DATA LPAR/1H(/,RPAR/1H)/,COMMA/1H,/,BLANK/1H /                   IO    11
      DATA (IALPH1(I),I=1,4)/1HR,1HE,1HA,1HD/                          IO    12
      DATA (IALPH2(I),I=1,5)/1HW,1HR,1HI,1HT,1HE/                      IO    13
      DATA (IALPH3(I),I=1,8)/1HC,1HO,1HN,1HT,1HI,1HN,1HU,1HE/          IO    14
      IFRMT=0                                                          IO    15
      IF(ITYP .EQ. 12) GO TO 10                                        IO    16
      DO 5 I=1,4                                                       IO    17
      IF(NEXT(JPTR) .NE. IALPH1(I)) GO TO 50                           IO    18
    5 CONTINUE                                                         IO    19
      GO TO 20                                                         IO    20
   10 DO 15 I=1,5                                                      IO    21
      IF(NEXT(JPTR) .NE. IALPH2(I)) GO TO 50                           IO    22
   15 CONTINUE                                                         IO    23
   20 IF(NEXT(JPTR) .NE. LPAR) GO TO 50                                IO    24
      CALL GNLE                                                        IO    25
      IF(JTYP .EQ. 2) GO TO 22                                         IO    26
      CALL ERROR(22,KDM1,KDM2)
      GO TO 28                                                         IO    28
   22 CALL SEARCH                                                      IO    29
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID,KDM2)
      IF(ISRCH(1) .EQ. 1) GO TO 25                                     IO    31
      IDTYP =1                                                         IO    32
      CALL STORE                                                       IO    33
      LOC=NID                                                          IO    34
   25 CALL IMPTYP                                                      IO    35
      IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4) CALL ERROR(22,KDM1,KDM2)
      IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 1) CALL ERROR(19,NXTID,KDM2)
      NBLOCK=NBLOCK+1                                                  IO    38
      IBLOCK(NBLOCK)=2000+LOC                                          IO    39
   28 IF(NEXT(JPTR) .EQ. COMMA) GO TO 40                               IO    40
      JPTR=JPTR-1                                                      IO    41
      GO TO 30                                                         IO    42
   40 CALL GNLE                                                        IO    43
      IF(JTYP .NE. 5) GO TO 26                                         IO    44
      CALL STSRCH                                                      IO    45
      STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,12)                         IO    46
      GO TO 29                                                         IO    47
   26 IF(JTYP .NE. 2) GO TO 50                                         IO    48
      CALL SEARCH                                                      IO    49
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID,KDM2)
      IF(ISRCH(1) .EQ. 1) GO TO 27                                     IO    51
      IDTYP=1                                                          IO    52
      CALL STORE                                                       IO    53
      LOC=NID                                                          IO    54
   27 CALL IMPTYP                                                      IO    55
      IF(BITGET(IDTBL(3,LOC),1,1) .NE. 1) CALL ERROR(43,KDM1,KDM2)
```

```
      29 IFRMT=1                                                        IO  57
      30 IF(NEXT(JPTR) .NE. RPAR) GO TO 50                             IO  58
         IF(NEXT(JPTR) .NE. BLANK) GO TO 35                            IO  59
         IF(ITYP .EQ. 12 .AND. IFRMT .EQ. 0) CALL ERROR(44,KDM1,KDM2)
         GO TO 36
      35 JPTR=JPTR-1                                                   IO  63
         CALL EXPR                                                     IO  64
         NTYPE=3                                                       IO  65
         CALL PARSE                                                    IO  66
         CALL IOSTR
      36 IF(MODE .NE. 1 .AND. ITYP .EQ. 11) GO TO 37
         RETURN
      37 IF(ITYPE(1) .EQ. 2) GO TO 42                                  IO  68
         A(1)=1HC                                                      IO  69
         RETURN                                                        IO  70
      42 WRITE(IOP,45) (A(I),I=1,6),(IALPH3(I),I=1,8)                  IO  71
      45 FORMAT(72A1)                                                  IO  72
         A(1)=1HC                                                      IO  73
         DO 47 I=2,6                                                   IO  74
      47 A(I)=1H                                                       IO  75
         RETURN                                                        IO  76
      50 CALL ERROR(7,KDM1,KDM2)
         RETURN                                                        IO  78
         END                                                           IO  79
```

```
         SUBROUTINE IOSTR
         COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),
        * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
        2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES
         COMMON/FUNC/IFNCRA(5,17),MARGS,IARGS(50),FNCLOC(5),NFUNC
         COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
         INTEGER BITGET
         DO 100 I=1,MARGS
         NBLOCK=NBLOCK+1
         ICOL=18*MOD(I-1,2)+9
         IVR=(I+1)/2
         LOC=BITGET(IARGS(IVR),ICOL,9)
         ISUB=BITGET(IARGS(IVR),ICOL+3,3)
         IF(ISUB .EQ. 1) GO TO 90
         IF(ISUB .EQ. 2) GO TO 20
         IF(ITYP .EQ. 11) GO TO 80
         IF(ITYP .EQ. 12) GO TO 90
      20 IDEF=BITGET(IARGS(IVR),ICOL+9,6)
         NMOVE=I-IDEF-1
         DO 30 J=1,NMOVE
         ITEMP=NBLOCK-J
      30 IBLOCK(ITEMP+1)=IBLOCK(ITEMP)
         IBLOCK(ITEMP)=1000+LOC
         NBLOCK=NBLOCK+1
         IBLOCK(NBLOCK)=6000+LOC
         GO TO 100
      80 IBLOCK(NBLOCK)=1000+LOC
         GO TO 100
      90 IBLOCK(NBLOCK)=2000+LOC
     100 CONTINUE
         RETURN
         END
```

```
      FUNCTION IPREV(IA)                                                IPREV  2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      INTEGER A,BLANK                                                   IPREV  4
      DATA BLANK/1H /                                                   IPREV  5
      DO 10 I=1,N                                                       IPREV  6
      J=IA-I+1                                                          IPREV  7
      IF(J .EQ. 0) GO TO 20                                             IPREV  8
      IF(A(J) .EQ. BLANK) GO TO 10                                      IPREV  9
      IPREV=ITYPE(J)                                                    IPREV 10
      JPTR=J-1                                                          IPREV 11
      RETURN                                                            IPREV 12
   10 CONTINUE                                                          IPREV 13
   20 IPREV=3                                                           IPREV 14
      JPTR=0                                                            IPREV 15
      RETURN                                                            IPREV 16
      END                                                               IPREV 17
```

```
      FUNCTION ITYPE(ID)                                                ITYPE  2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      INTEGER BITGET                                                    ITYPE  4
      NXT=NEXT(ID)                                                      ITYPE  5
      IVAL=BITGET(NXT,6,6)                                              ITYPE  6
C     IF UNIVAC 1108 - PLACE A 'C' IN COLUMN 1 OF NEXT TWO CARDS        CH34  25
C     IF( IVAL .GE. IRA .AND. IVAL .LE. IRZ) GO TO 10
C     IF(IVAL .GE. IR0 .AND. IVAL .LE. IR9) GO TO 20
C     IF CDC 6700 - PLACE A 'C' IN COLUMN 1 OF NEXT TWO CARDS           CH34  26
      IF(IVAL .GE. 6 .AND. IVAL .LE. 31) GO TO 10
      IF(IVAL .GE. 48 .AND. IVAL .LE. 57) GO TO 20
      ITYPE=3                                                           ITYPE  9
      RETURN                                                            ITYPE 10
   10 ITYPE=1                                                           ITYPE 11
      RETURN                                                            ITYPE 12
   20 ITYPE=2                                                           ITYPE 13
      RETURN                                                            ITYPE 14
      END                                                               ITYPE 15
```

```
      SUBROUTINE LOGCHK                                                 LOGCHK 2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      COMMON/LOGIC/LOG,LOGST                                            LOGCHK 4
      DIMENSION LOGOP(11),LOGRA(5)                                      LOGCHK 5
      DATA (LOGOP(I),I=1,11)/2HLT,2HLE,2HGT,2HGE,2HEQ,2HNE,2HOR,3HAND, LOGCHK 6
     * 3HNOT,4HTRUE,5HFALSE/                                            LOGCHK 7
      JPTR=LOGST                                                        LOGCHK 8
      DO 10 I=1,6                                                       LOGCHK 9
      NXT=NEXT(JPTR)                                                    LOGCHK10
      IF(NXT .EQ. 1H.) GO TO 12                                         LOGCHK11
   10 LOGRA(I)=NXT                                                      LOGCHK12
      GO TO 20                                                          LOGCHK13
   12 IF(I .LT. 3) GO TO 20                                             LOGCHK14
      CALL CAA(LOGRA,I-1,LOG)                                           LOGCHK15
      DO 15 I=1,11                                                      LOGCHK16
      IF(LOG .EQ. LOGOP(I)) GO TO 30                                    LOGCHK17
   15 CONTINUE                                                          LOGCHK18
   20 LOG=0                                                             LOGCHK19
      RETURN                                                            LOGCHK20
   30 LOG=I                                                             LOGCHK21
      LOGID=I                                                           LOGCHK22
      RETURN                                                            LOGCHK23
      END                                                               LOGCHK24
```

```
      SUBROUTINE LOGIF                                                  LOGIF  2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      COMMON/STRING/NTYPE,NSTR,STR(500)                                 LOGIF  4
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      INTEGER A,STR,BLANK,AY,EF                                         LOGIF  6
      DATA LPAR/1H(/,BLANK/1H /,AY/1HI/,EF/1HF/                         LOGIF  7
      IF(NEXT(JPTR) .NE. AY) GO TO 110                                  LOGIF  8
      IF(NEXT(JPTR) .NE. EF) GO TO 110                                  LOGIF  9
      IF(NEXT(JPTR) .NE. LPAR) GO TO 110                                LOGIF 10
      JPTR=JPTR-1                                                       LOGIF 11
      CALL EXPR                                                         LOGIF 12
      NSTR=NSTR+1                                                       LOGIF 13
      STR(NSTR)= -5                                                     LOGIF 14
      NTYPE=2                                                           LOGIF 15
      CALL PARSE                                                        LOGIF 16
      CALL FNCSTR                                                       LOGIF 17
      CALL BLKSTR
      IF(ITYP .GT. 15) GO TO 130                                        LOGIF 18
      LTYP=1                                                            LOGIF 19
      GO TO (10,20,30,40,50,60,70,80,70,70,90,90,100,100,100),ITYP      LOGIF 20
   10 CALL INIT                                                         LOGIF 21
      RETURN                                                            LOGIF 22
   20 CALL ASSIGN                                                       LOGIF 23
      RETURN                                                            LOGIF 24
   30 CALL GOTO                                                         LOGIF 25
      NBLOCK=NBLOCK+1                                                   LOGIF 26
      IBLOCK(NBLOCK)=998                                                LOGIF 27
      NBRNCH=2                                                          LOGIF 28
      RETURN                                                            LOGIF 29
   40 CALL ASGOTO                                                       LOGIF 30
      NBLOCK=NBLOCK+1                                                   LOGIF 31
      IBLOCK(NBLOCK)=998                                                LOGIF 32
      NBRNCH=NBRNCH+1                                                   LOGIF 33
      RETURN                                                            LOGIF 34
   50 CALL CTGOTO                                                       LOGIF 35
      NBLOCK=NBLOCK+1                                                   LOGIF 36
      IBLOCK(NBLOCK)=998                                                LOGIF 37
      NBRNCH=NBRNCH+1                                                   LOGIF 38
      RETURN                                                            LOGIF 39
   60 CALL ARIF                                                         LOGIF 40
      NBLOCK=NBLOCK+1                                                   LOGIF 41
      IBLOCK(NBLOCK)=998                                                LOGIF 42
      NBRNCH=NBRNCH+1                                                   LOGIF 43
      RETURN                                                            LOGIF 44
   70 CALL SIMP                                                         LOGIF 45
      IF(ITYP .EQ. 7) RETURN                                            LOGIF 46
      NBLOCK=NBLOCK+1                                                   LOGIF 47
      IBLOCK(NBLOCK)=998                                                LOGIF 48
      NBRNCH=2                                                          LOGIF 49
      RETURN                                                            LOGIF 50
   80 CALL CALL                                                         LOGIF 51
      RETURN                                                            LOGIF 52
   90 CALL IO                                                           LOGIF 53
      RETURN                                                            LOGIF 54
  100 CALL AUXIO                                                        LOGIF 55
      RETURN                                                            LOGIF 56
  110 CALL ERROR(7,KDM1,KDM2)
      RETURN                                                            LOGIF 58
  130 CALL ERROR(45,KDM1,KDM2)
      RETURN                                                            LOGIF 60
      END                                                               LOGIF 61
```

```
      SUBROUTINE LOOPCK                                                 LOOPCK 2
      COMMON/LABELS/STATRA(2,200),NLABEL                                LOOPCK 3
      COMMON/DOLOOP/ISTACK(4,50),NSTACK,ILOOP                           LOOPCK 4
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      INTEGER STATRA,BITGET                                             LOOPCK 6
      IF(NSTACK .EQ. 0) RETURN                                          LOOPCK 7
      IBLKST=1                                                          LOOPCK 8
   10 IBLKND=BITGET(IBLOCK(IBLKST),28,16)-1                             LOOPCK 9
      LOOPZ=BITGET(IBLOCK(IBLKST),12,6)                                 LOOPCK10
      NBR=BITGET(IBLOCK(IBLKST),6,6)                                    LOOPCK11
      IF(IBLKND .EQ. -1) IBLKND=NBLOCK                                  LOOPCK12
      IST=IBLKND-NBR+1                                                  LOOPCK13
      DO 100 I=IST,IBLKND                                               LOOPCK14
      JLOOP=LOOPZ                                                       LOOPCK15
      IF(IBLOCK(I) .GE. 998) GO TO 100                                  LOOPCK16
      IBLK=IBLOCK(I)
      NXTBLK=BITGET(STATRA(2,IBLK),36,18)
      KLOOP=BITGET(IBLOCK(NXTBLK),12,6)                                 LOOPCK18
      IF(KLOOP .EQ. 0) GO TO 100                                        LOOPCK19
      IF(JLOOP .EQ. 0) GO TO 200                                        LOOPCK20
   50 IF(JLOOP .EQ. KLOOP) GO TO 100                                    LOOPCK21
      JLOOP=ISTACK(3,JLOOP)                                             LOOPCK22
      IF(JLOOP .EQ. 0) GO TO 200                                        LOOPCK23
      GO TO 50                                                          LOOPCK24
  200 IBLK=IBLOCK(I)
      WRITE(6,201) STATRA(1,IBLK)
  201 FORMAT(6X,65H ILLEGAL TRANSFER INTO THE RANGE OF A DO LOOP AT STATLOOPCK26
     1EMENT NUMBER,I6)                                                  LOOPCK27
  100 CONTINUE                                                          LOOPCK28
      IF(IBLKND .EQ. NBLOCK) RETURN                                     LOOPCK29
      IBLKST=IBLKND+1                                                   LOOPCK30
      GO TO 10                                                          LOOPCK31
      END                                                               LOOPCK32
```

```
      SUBROUTINE LVDLET                                                 IZ05177
      COMMON/LVARGS/IFUNC,IARG,IADD,IPOS,ITYP,IVAL,LSTHED,NVAL,
     +  IDSTRY,IVALS(10),ITYPI(10),NSKIP                               IZ05177
      INTEGER FLGSPC,FLOMSK,FL1MSK,FL2MSK,FL5MSK,FLG67,REGASP,THIS      IZ08251
     +  ,FL3MSK,FL4MSK,SEQSPC                                          COMPAND
      COMMON/LVVTR1/MEMSZE,REGASP,NODSPC(    1)/LVVTR2/LSTSPC(    1)/  GIRDLET
     +LVVTR3/LNKSPC(    1)/LVVTR4/FLGSPC(    1)                       GIRDLET
      COMMON/LVFLAG/FLOMSK,FL1MSK,FL2MSK,FL3MSK,FL4MSK,FL5MSK,FLG67     IZ05177
      COMMON /LVTABL/ MAPSZE,MAP(1) /LVVSEQ/ ISEQSZ,SEQSPC(1)           IZ05177
      DATA NFLG02/95/                                                   RR1
      KONFLC=0                                                          GIRDLET
C     DETERMINE DIRECTION TO PROCEED FOR MULTIVALUE LISTS               IZ04127
      JPOS=IPOS
      IPOS=IABS(IPOS)                                                   IZ05032
      IF(IADD.NE.-1)  GO TO  74                                         GIRDLET
      IF(IARG.EQ.-1) GO TO 66                                           GIRDLET
      IADD=IFUNC+IARG                                                   IZ07261
      IF(IADD.GT.MEMSZE) IADD=IADD-MEMSZE                               IZ07261
      ILOG=FLGSPC(IADD) .AND. FL5MSK
      IF(ILOG .EQ. 0) GO TO 99
    1 IF(NODSPC(IADD).EQ.IARG) GO TO 4                                  GIRDLET
C     SEARCH CONFLICT LIST FOR THE FUNCTION                             IZ04127
      IADD=LNKSPC(IADD)                                                 GIRDLET
      ILOG=FLGSPC(IADD) .AND. FL5MSK
      IF(ILOG .NE. 0) GO TO 99
      GO TO 1                                                           GIRDLET
   66 IADD=IFUNC                                                        GIRDLET
C     TO DELETE A SPECIFIC TYPE OF NODE (INDEXED DELETE), GO TO 72      IZ04127
    4 IF(ITYP.NE.-1) GO TO 72                                           GIRDLET
      ILOG=FLGSPC(IADD) .AND. FLOMSK
      IF(ILOG .EQ. 0) GO TO 6
      ISADD=LSTSPC(IADD)                                                GIRDLET
C     DELETE ENTIRE MULTIVALUED FUNCTION, RETRIEVE FIRST VALUE          IZ04127
      IVAL=NODSPC(ISADD)                                                IZ04127
    5 NXTADD=LSTSPC(ISADD)                                              GIRDLET
      NODSPC(ISADD)=NODSPC(REGASP)                                      GIRDLET
      LSTSPC(ISADD)=REGASP                                              GIRDLET
      LNKSPC(ISADD)=0                                                   GIRDLET
      FLGSPC(ISADD)=0                                                   IZ08251
      ISUB=NODSPC(REGASP)
      LSTSPC(ISUB)=ISADD
      NODSPC(REGASP)=ISADD                                              GIRDLET
      ILOG=FLGSPC(NXTADD) .AND. FLOMSK
      IF(ILOG .NE. 0) GO TO 2
      ISADD=NXTADD                                                      GIRDLET
      GO TO 5                                                           GIRDLET
C     FUNCTION IS SINGLE VALUED, RETRIEVE VALUE                         IZ04127
    6 IVAL=LSTSPC(IADD)                                                 IZ04127
    2 ILOG=FLGSPC(IADD) .AND. FL5MSK
      IF(ILOG .EQ. 0) GO TO 68
      NXFUNC=LNKSPC(IADD)                                               GIRDLET
      ILOG=FLGSPC(NXFUNC) .AND. FL5MSK
      IF(ILOG .NE. 0) GO TO 10
      NODSPC(IADD)=NODSPC(NXFUNC)                                       GIRDLET
      LSTSPC(IADD)=LSTSPC(NXFUNC)                                       GIRDLET
      LNKSPC(IADD)=LNKSPC(NXFUNC)                                       GIRDLET
      FLGSPC(IADD)=FLGSPC(NXFUNC)                                       IZ08251
      FLGSPC(IADD)=FLGSPC(IADD).OR.FL5MSK                               GIRDLET
      ILOG=FLGSPC(IADD) .AND. FLOMSK
      IF(ILOG .EQ. 0) GO TO 9
      KVAL=LSTSPC(IADD)                                                 IZ12021
    8 KVAL=LSTSPC(KVAL)                                                 GIRDLET
      ISUB=LSTSPC(KVAL)
      ILOG=FLGSPC(ISUB) .AND. FLOMSK
      IF(ILOG .EQ. 0) GO TO 8
      LSTSPC(KVAL)=IADD                                                 GIRDLET
    9 IADD=NXFUNC                                                       GIRDLET
   10 NODSPC(IADD)=NODSPC(REGASP)                                       GIRDLET
```

```
      11  LSTSPC(IADD)=REGASP                                          GIRDLET
          LNKSPC(IADD)=0                                               GIRDLET
          FLGSPC(IADD)=0                                               IZ08251
          ISUB=LSTSPC(IADD)
          NODSPC(ISUB)=IADD
          ISUB=NODSPC(IADD)
          LSTSPC(ISUB)=IADD
          RETURN                                                       GIRDLET
      72  ILOG=FLGSPC(IADD) .AND. FLOMSK
          IF(ILOG .NE. 0) GO TO 20
          IF(IPOS.NE.1) GO TO 99                                       IZ05052
          IF(ITYP.EQ.3) GO TO 6                                        IZ04272
          ISTYP=(FLGSPC(IADD).AND.FLG67)                               IZ04272
          IF(ISTYP.EQ.ITYP) GO TO 6                                    GIRDLET
      99  IVAL=-1                                                     IZ07287
          RETURN                                                       GIRDLET
      20  IND=0                                                        GIRDLET
          FLGSPC(IADD)=FLGSPC(IADD).OR.FL4MSK                          COMPAND
          LAST=IADD                                                    GIRDLET
          IF(JPOS)121,99,21                                            IZ04272
     121  ISUB=LSTSPC(IADD)
          LASTI=LNKSPC(ISUB)
          THIS=LASTI                                                   IZ04272
          GO TO 27                                                     IZ04272
      21  IF(JPOS.LT.0) GO TO 80                                       IZ04272
          THIS=LSTSPC(LAST)                                            IZ04272
          ILOG=FLGSPC(THIS) .AND. FLOMSK
          IF(ILOG .NE. 0) GO TO 99
          GO TO 27                                                     IZ04272
      80  THIS=LNKSPC(LAST)                                           IZ04272
          IF(THIS.EQ.LASTI) GO TO 99                                   IZ04272
      27  IF(ITYP.EQ.3) GO TO 23                                       IZ04272
          ISTYP=(FLGSPC(THIS).AND.FLG67)                               IZ04272
          IF(ISTYP.EQ.ITYP)   GO TO 23                                 GIRDLET
      22  LAST=THIS                                                    GIRDLET
          GO TO 21                                                     GIRDLET
      23  IND=IND+1                                                    GIRDLET
          IF(IND.NE.IPOS)     GO TO 22                                 IZ02092
C     RETRIEVE THE IPOS@TH OF THE KTYP@TH VALUE BEFORE DELETING        IZ04127
          IVAL=NODSPC(THIS)                                            IZ04127
          MADD=IADD                                                    IZ05032
          IF(JPOS.GT.0) GO TO 55                                       IZ05032
          NEXT=LNKSPC(THIS)                                            IZ05032
          IF(THIS.EQ.LASTI) GO TO 82                                   IZ05032
          LNKSPC(LAST)=NEXT                                            IZ05032
          GO TO 83                                                     IZ05032
      82  ISUB=LSTSPC(IADD)
          LNKSPC(ISUB)=NEXT
      83  IF(NEXT.EQ.LASTI) GO TO 84                                   IZ05032
          LSTSPC(NEXT)=LAST                                            IZ05032
          GO TO 85                                                     IZ05032
      84  LSTSPC(IADD)=LAST                                            IZ05032
      85  IADD=THIS                                                    IZ05032
          GO TO 86                                                     IZ05032
      55  NEXT=LSTSPC(THIS)                                           IZ05032
          ILOG=FLGSPC(NEXT) .AND. FLOMSK
          IF( ILOG .NE. 0) GO TO 50
          LNKSPC(NEXT)=LAST                                            IZ04272
```

```
127*            GO TO 24                                                IZ04272
128*         50 ISUB=LSTSPC(IADD)
129*            LNKSPC(ISUB)=LAST
130*         24 IADD=THIS                                               IZ02092
131*            JLAST=LNKSPC(THIS)                                      IZ02092
132*            ISUB=LSTSPC(JLAST)
133*            ILOG=FLGSPC(ISUB) .AND. FLOMSK
134*            IF(ILOG .NE. 0) LNKSPC(NEXT)=JLAST
135*            LSTSPC(LAST)=NEXT                                       IZ03032
136*         86 KLAST=LSTSPC(MADD)                                      IZ05032
137*            IF(LNKSPC(KLAST).NE.KLAST) GO TO 10                     IZ02292
138*     C                                                              IZ02292
139*     C      CONVERT TO SINGLE VALUE LIST                            IZ02292
140*     C                                                              IZ02292
141*            LSTSPC(MADD)=NODSPC(KLAST)                              IZ02292
142*            ILOG=FLGSPC(MADD) .OR. FLGSPC(KLAST)
143*            FLGSPC(MADD)=ILOG .AND. NFLG02
144*            FLGSPC(KLAST)=0                                         IZ02292
145*            LNKSPC(KLAST)=0                                         IZ02292
146*            NODSPC(KLAST)=NODSPC(REGASP)                            IZ02292
147*            LSTSPC(KLAST)=REGASP                                    IZ02292
148*            ISUB=LSTSPC(KLAST)
149*            NODSPC(ISUB)=KLAST
150*            ISUB=NODSPC(KLAST)
151*            LSTSPC(ISUB)=KLAST
152*            GO TO 10                                                GIRDLET
153*         74 ILOG=FLGSPC(IADD) .AND. FLOMSK
154*            IF(ILOG .NE. 0) GO TO 99
155*            ILOG=FLGSPC(IADD) .AND. FL2MSK
156*            IF(ILOG .EQ. 0) GO TO 99
157*            LAST=LNKSPC(IADD)                                       GIRDLET
158*            NEXT=LSTSPC(IADD)                                       GIRDLET
159*            LSTSPC(LAST)=NEXT                                       GIRDLET
160*            ILOG=FLGSPC(NEXT) .AND. FLOMSK
161*            IF(ILOG .NE. 0) GO TO 10
162*            LNKSPC(NEXT)=LAST                                       GIRDLET
163*            GO TO 10                                                GIRDLET
164*         68 NEXT=LNKSPC(IADD)                                       GIRDLET
165*            NEXTI=NEXT                                              GIRDLET
166*         25 IF(LNKSPC(NEXTI).EQ.IADD)  GO TO 26                     GIRDLET
167*            NEXTI=LNKSPC(NEXTI)                                     GIRDLET
168*            GO TO 25                                                GIRDLET
169*         26 LNKSPC(NEXTI)=NEXT                                      GIRDLET
170*            KONFLC=1                                                GIRDLET
171*            GO TO 10                                                GIRDLET
172*            END                                                     GIRDLET
```

```
      SUBROUTINE LVEXIT(N)
      COMMON/LVARGS/LVFUNC,LVVARG,LVVAD,LVVPOS,LVVTYP,LVVAL,
     +LVHEAD,LVVNVL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP
      COMMON/LVTABL/LVTSIZ,LVMAP(     1)/LVVSEQ/LVSIZE,LVSQSP(     1)
      COMMON /TYP/ NN(3),ERRFLG
      COMMON /STRING/ NTYPE,NSTR
      COMMON /NEED/ START,ASSOC,LEVEL,STOP
      COMMON/NEEDS/STJ,JSTACK,R,JAS,J,JLAST,RTEMP,STACK(400,4)
      INTEGER R,RTEMP,STJ,STACK,ASSOC,START,STOP
      COMMON /GIRL/ MM(19),OPRAND
      COMMON /HL/ HOL,ACTION,FUNC1,FUNC2,FUNC3,LEFT,RIGHT,STRING,MAXJ
      COMMON /NTIMES/ NTIMES,MAXI
      INTEGER STRING,HOL,ACTION,RIGHT,FUNC1,FUNC2,FUNC3,OPRAND
      LOGICAL ERRFLG
C     EXECUTE
      GO TO 25000
25001 CONTINUE
      IF(MAXJ .NE. 0) PRINT 100,MAXJ
  100 FORMAT(1X,44H STATEMENT IS TOO COMPLEX. CORRECT TO CHAR. ,I3)
      IF(MAXJ .EQ. 0) PRINT 200,MAXI,NSTR
  200 FORMAT(1X,29H STATEMENT TOO LONG AT CHAR. ,I3,3H OF,I3)
C COMMENTS USED IN CASE OF GIRS PROBLEMS WHEN MEMORY USED UP
C     GO TO 10
C     IF(MAXJ .EQ. 0) GO TO 50
      IF(MAXJ .EQ. 0) GO TO 10
      DO 30 NCHAR=1,MAXJ
C     STRING=HOL.NCHAR(-LEFT,-RIGHT,-HOL,-STRING)
C****      STRING            =           HOL
      LVVPOS =       NCHAR
      LVVTYP =    3
      LVFUNC=          HOL
      LVVARG=       STRING
      CALL LVFIND(LV2       A,LV2       B,LV2       C,LV2       D)
      LVIAAD=STRING
      IF(LVVAL .NE. -1) LVIAAD=LVVAL
      LVIAAI=LVIAAD
C**** LVI     AAX             -          LEFT
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=       LEFT
      LVVARG=LVIAAI
      CALL LVDLET
      LVIAAI=LVIAAD
C**** LVI     AAX             -         RIGHT
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=      RIGHT
      LVVARG=LVIAAI
      CALL LVDLET
      LVIAAI=LVIAAD
C**** LVI     AAX             -           HOL
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=         HOL
      LVVARG=LVIAAI
      CALL LVDLET
      LVIAAI=LVIAAD
C**** LVI     AAX             -        STRING
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
```

```
      LVFUNC=    STRING
      LVVARG=LV1AA1
      CALL LVDLET
   30 CONTINUE
C     STRING-STRING
C**** STRING                -       STRING
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    STRING
      LVVARG=    STRING
      CALL LVDLET
C     OPRAND(-OPRAND,-STRING,-ACTION,-FUNC1)
      LV1AAD=OPRAND
C**** LV1    AAT            -       OPRAND
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    OPRAND
      LVVARG=LV1AAD
      CALL LVDLET
C**** LV1    AAT            -       STRING
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    STRING
      LVVARG=LV1AAD
      CALL LVDLET
C**** LV1    AAT            -       ACTION
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    ACTION
      LVVARG=LV1AAD
      CALL LVDLET
C**** LV1    AAT            -       FUNC1
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    FUNC1
      LVVARG=LV1AAD
      CALL LVDLET
   10 CONTINUE
      REWIND 19
      NTIMES=0
C  10 CONTINUE
C     REWIND 99
C     NTIMES=0
      J=NSTR+1
      R=STOP
      STJ=R
      ERRFLG=.TRUE.
      JSTACK=1
      STACK(JSTACK,1)=STOP
      STACK(JSTACK,2)=100
      STACK(JSTACK,3)=J
      STACK(JSTACK,4)=0
C     NSTR=0
      NSTR=MAXJ
C     COMPLETE
      RETURN
25000 CONTINUE
      LV2A=0
      LV2B=0
      LV2C=0
      LV2D=0
      LV2E=0
      LV2F=0
      LV2G=0
      LV2H=0
      GO TO 25001
      END
```

```
      SUBROUTINE LVFECH(N)                                                  IZ05177
      INTEGER FLGSPC,SEQSPC,REGASP                                          COMPAND
      COMMON /LVTABL/ MAPSZE,MAP(1) /LVVSEQ/ ISEQSZ,SEQSPC(1)               IZ05177
      COMMON/LVVTR1/MEMSZE,REGASP,NODSPC(    1)/LVVTR2/LSTSPC(    1)/     IZ07081
     *LVVTR3/LNKSPC(    1)/LVVTR4/FLGSPC(    1)                           IZ07081
      COMMON/LVRAND/KPRIME,KS,KX,KDY,KDX,KTEMP
      READ(N) MEMSZE,REGASP,KPRIME,KS,KX,KTEST,KDY,KTEMP,KDX,KNUM           IZ02147
     *,ISEQSZ                                                               IZ05177
      READ(N)(NODSPC(I),I=1,MEMSZE)                                         IZ07081
      READ(N)(LSTSPC(I),I=1,MEMSZE)                                         IZ07081
      READ(N)(LNKSPC(I),I=1,MEMSZE)                                         IZ07081
      READ(N)(FLGSPC(I),I=1,MEMSZE)                                         IZ07081
      READ(N)(SEQSPC(I),I=1,ISEQSZ)                                         IZ05177
      PRINT 10                                                              COMPAND
   10 FORMAT(////1X,34H GRAPH HAS BEEN PLACED INTO MEMORY///)
      RETURN                                                                IZ07081
      END                                                                   IZ07081
```

```
      SUBROUTINE LVFIND(INDEX,INDXAD,KFUNC,KARG)                        IZ05177
      COMMON/LVARGS/IFUNC,IARG,IADD,IPOS,ITYP,IVAL,LSTHED,NVAL,
     +  IDSTRY,IVALS(10),ITYP1(10),NSKIP                               IZ05177
      INTEGER FLGSPC,REGASP,FL0MSK,FL1MSK,FL2MSK,FL3MSK,FL4MSK,FL5MSK,  COMPAND
     + FLG67,SEQSPC                                                     COMPAND
      COMMON/LVFLAG/FL0MSK,FL1MSK,FL2MSK,FL3MSK,FL4MSK,FL5MSK,FLG67     IZ05177
      COMMON /LVTABL/ MAPSZE,MAP(1) /LVVSEQ/ ISEQSZ,SEQSPC(1)           IZ05177
      COMMON/LVVTR1/MEMSZE,REGASP,NODSPC(    1)/LVVTR2/LSTSPC(    1)/  GIRFIND
     *LVVTR3/LNKSPC(    1)/LVVTR4/FLGSPC(    1)                        GIRFIND
      DATA NFLAG4/247/                                                  RR1
      IADD=IFUNC+IARG                                                   IZ07261
      IF(IADD.GT.MEMSZE) IADD=IADD-MEMSZE                               IZ07261
      LSTHED=0                                                          GIRFIND
      ILOG=FLGSPC(IADD) .AND. FL5MSK
      IF(ILOG .EQ. 0) GO TO 99
    1 IF(NODSPC(IADD).EQ.IARG)  GO TO 4                                 GIRFIND
      IADD=LNKSPC(IADD)                                                 GIRFIND
      ILOG=FLGSPC(IADD) .AND. FL5MSK
      IF(ILOG .NE. 0) GO TO 99
      GO TO 1                                                           GIRFIND
    4 ILOG=FLGSPC(IADD) .AND. FL0MSK
      IF(ILOG .NE. 0) GO TO 14
      ISTYP=(FLGSPC(IADD).AND.FLG67)                                    GIRFIND
      IF(ITYP.EQ.3) GO TO 11                                            GIRFIND
      IF(ISTYP.EQ.3)ISTYP=2                                             GIRFIND
      IF(ISTYP.NE.ITYP)  GO TO 99                                       IZ01042
   11 IVAL=LSTSPC(IADD)                                                 GIRFIND
      IF((IPOS.NE.1).AND.(IPOS.NE.-1))  GO TO 99                        IZ02042
      ITYP=(FLGSPC(IADD).AND.FLG67)                                     GIRFIND
      LSTHED=-1                                                         GIRFIND
      RETURN                                                            GIRFIND
   14 LSTHED=IADD                                                       GIRFIND
      IND=0                                                             IZ12271
      KNDEX=IABS(INDEX)                                                 IZ12271
      JPOS=IABS(IPOS)                                                   IZ12271
      IF(NSKIP.EQ.1) GO TO 50                                           COMPAND
      IF((KFUNC.NE.IFUNC).OR.(KARG.NE.IARG)) GO TO 50                   IZ01107
      ILOG=FLGSPC(LSTHED) .AND. FL4MSK
      IF(ILOG .NE. 0) GO TO 50
      IF((IPOS*INDEX) .LE. 0) GO TO 50
      IF(JPOS.LT.2) GO TO 50                                            IZ08237
      NDX=FLGSPC(INDXAD)                                                IZ05177
      ILOG=NDX .AND. FL5MSK
      IF(ILOG .NE. 0) GO TO 50
      ILOG=NDX .AND. FL1MSK
      IF(ILOG .EQ. 0) GO TO 50
      IF(JPOS.GE.KNDEX)  GO TO 25                                       IZ12271
      IF((JPOS+JPOS).LE.KNDEX)  GO TO 50                                IZ12271
      IF(IPOS) 30,99,40
   50 FLGSPC(LSTHED)=FLGSPC(LSTHED).AND.NFLAG4                          COMPAND
      IF(IPOS) 20,99,10
C                                                                       IZ12271
COUNT DOWN FROM THE TOP  OF THE LIST                                    IZ12271
C                                                                       IZ12271
   10 IADD=LSTSPC(IADD)                                                 IZ12271
      ILOG=FLGSPC(IADD) .AND. FL0MSK
      IF(ILOG .NE. 0) GO TO 99
      ISTYP=(FLGSPC(IADD).AND.FLG67)                                    IZ12271
      IF(ITYP.EQ.3) GO TO 22                                            IZ12271
      IF(ISTYP.EQ.3)ISTYP=2                                             IZ12271
```

```
61•             IF(ISTYP.NE.ITYP) GO TO 10                                  IZ12271
62•          22 IND=IND+1                                                   IZ12271
63•             IF(IND.NE.JPOS) GO TO 10                                    IZ12271
64•          28 IVAL=NODSPC(IADD)                                           IZ12271
65•             ITYP=(FLGSPC(IADD).AND.FLG67)                               IZ12271
66•           55 INDEX=IPOS
67•             INDXAD=IADD                                                 IZ05177
68•             KFUNC=IFUNC                                                 IZ01107
69•             KARG=IARG                                                   IZ01107
70•             RETURN                                                      IZ12271
71•      C                                                                  IZ12271
72•      COUNT UP FROM THE BOTTOM OF THE LIST                               IZ12271
73•      C                                                                  IZ12271
74•          20 IADD=LSTSPC(IADD)                                           IZ12271
75•             KTEST=0                                                     IZ01042
76•          23 IADD=LNKSPC(IADD)                                           IZ12271
77•             IF(KTEST.EQ.0) GO TO 24                                     IZ01042
78•             ISUB=LSTSPC(IADD)
79•             ILOG=FLGSPC(ISUB) .AND. FLGMSK
80•             IF(ILOG .NE. 0) GO TO 99
81•          24 KTEST=1                                                     IZ01042
82•             ISTYP=(FLGSPC(IADD).AND.FLG67)                              IZ12271
83•             IF(ITYP.EQ.3) GO TO 21                                      IZ12271
84•             IF(ISTYP.EQ.3)ISTYP=2                                       IZ12271
85•             IF(ISTYP.NE.ITYP) GO TO 23                                  IZ12271
86•          21 IND=IND+1                                                   IZ12271
87•             IF(IND.NE.JPOS) GO TO 23                                    IZ12271
88•          29 IVAL=NODSPC(IADD)                                           IZ12271
89•             ITYP=(FLGSPC(IADD).AND.FLG67)                               IZ12271
90•             GO TO 55                                                    IZ12271
91•           25 IF(IPOS) 40,99,30
92•      C                                                                  IZ12271
93•      COUNT DOWN FROM INDXADD                                            IZ12271
94•      C                                                                  IZ12271
95•          30 JPOS=IABS(JPOS-KNDEX)                                       IZ08037
96•             IADD=INDXAD                                                 IZ05177
97•             IF(JPOS.EQ.0) GO TO 28                                      IZ12271
98•             GO TO 10                                                    IZ12271
99•      C                                                                  IZ12271
100•     COUNT UP FROM INDXADD                                              IZ12271
101•     C                                                                  IZ12271
102•         40 JPOS=IABS(JPOS-KNDEX)                                       IZ08037
103•            IADD=INDXAD                                                 IZ05177
104•            IF(JPOS.EQ.0) GO TO 29                                      IZ12271
105•            KTEST=1
106•            GO TO 23                                                    IZ12271
107•         99 IVAL=-1                                                     GIRFIND
108•            INDEX=0                                                     RR1
109•            INDXAD=0                                                    RR1
110•            KFUNC=0                                                     RR1
111•            KARG=0                                                      RR1
112•            RETURN                                                      GIRFIND
113•            END                                                         GIRFIND
```

```
      SUBROUTINE LVGRN(NODE)
      INTEGER FLGSPC,REGASP                                              IZ08251
      COMMON/LVVTR1/MEMSZE,REGASP,NODSPC(    1)/LVVTR2/LSTSPC(    1)/   IZ06161
     *LVVTR3/LNKSPC(    1)/LVVTR4/FLGSPC(    1)                         IZ06161
      COMMON/LVRAND/KPRIME,KSEED,NRO#,KDNODE,KDRO#,KTEMP
      NODE=KTEMP+KDNODE
      KTEMP=NODE
      KDNODE=KDNODE+1
      IF(NODE .GT. MEMSZE) GO TO 5
      RETURN
    5 IF(NRO# .GT. KPRIME) GO TO 10
      NRO#=NRO#+KSEED
      IF(NRO# .GT. KPRIME) NRO#=NRO#-KPRIME
      NODE=NRO#
      KTEMP=NODE
      KDNODE=KPRIME+1
      IF(NODE .NE. KSEED) RETURN
      NRO#=0
      KDRO#=KPRIME
   10 KDRO#=KDRO#+1
      NRO#=NRO#+KDRO#
      NODE=NRO#
      KTEMP=NODE
      KDNODE=KDRO#
      IF(NODE .GT. MEMSZE) GO TO 20
      RETURN                                                             IZ06161
   20 PRINT 15                                                           COMPAND
   15 FORMAT(1H ,1X,47H ERROR..,NUMBER OF NODES EXCEEDS REQUEST MEMORY/   RRI
     *27H THE PROGRAM IS TERMINATED.)                                    RRI
      STOP                                                               IZ06161
      END                                                                IZ06161
```

```
      SUBROUTINE LVNSRT                                                 IZ05177
      COMMON/LVARGS/IFUNC,IARG,IADD,IPOS,ITYP2,IVAL,LSTHED,NVAL,
     +  IDSTRY,IVALS(10),ITYP(10),NSKIP                                IZ05177
      INTEGER FLGSPC,FL0MSK,FL1MSK,FL2MSK,FL5MSK,FLG67,REGASP,TEMP,THIS, IZ08251
     + FLGTMP,TWO,THREE,HEAD,OLDLOC,ASPREG,SEQSPC,FL3MSK,FL4MSK         COMPAND
      COMMON/LVVTR1/MEMSZE,REGASP,NODSPC(    1)/LVVTR2/LSTSPC(    1)/ GIRNSRT
     +LVVTR3/LNKSPC(    1)/LVVTR4/FLGSPC(    1)                       GIRNSRT
      COMMON/LVFLAG/FL0MSK,FL1MSK,FL2MSK,FL3MSK,FL4MSK,FL5MSK,FLG67     IZ05177
      COMMON /LVTABL/ MAPSZE,MAP(1) /LVVSEQ/ ISEQSZ,SEQSPC(1)           IZ05177
      DATA TWO/2/,THREE/3/,NFLG67/252/                                  RRI
C                                                                       IZ06097
      FLGTMP=FL1MSK                                                     IZ06097
C                                                                       IZ03302
      IF(REGASP.EQ.LSTSPC(REGASP)) GO TO 98                             IZ03302
C                                                                       GIRNSRT
C   FORM FIRST WORD OF SINGLE OR MULTIVALUED FUNCTION                   GIRNSRT
      IF(NVAL.EQ.1)GO TO 20                                             GIRNSRT
      LSTTMP=REGASP                                                     GIRNSRT
      FLGTMP=(FLGTMP.OR.FL2MSK)                                         GIRNSRT
      FLGTMP=(FLGTMP.OR.FL0MSK)                                         GIRNSRT
      GO TO 21                                                          GIRNSRT
   20 LSTTMP=IVALS(1)                                                   GIRNSRT
   21 FLGTMP=(FLGTMP.OR.ITYP(1))                                        GIRNSRT
C                                                                       GIRNSRT
C---------------------------------------------------------------------- GIRNSRT
C-DETERMINE ADDRESS FOR FUNCTION                                        GIRNSRT
      IADD=IFUNC+IARG                                                   GIRNSRT
      IF(IADD.GT.MEMSZE) IADD=IADD-MEMSZE                               IZ07261
C                                                                       GIRNSRT
C   IF THAT ADDRESS IS ALREADY IN WORKING SPACE, GO TO 25               GIRNSRT
      IF(IDSTRY-1) 125,300,350                                          IZ05177
  125 ILOG=FL1MSK .AND. FLGSPC(IADD)
      IF((ILOG .NE. 0) GO TO 25
C                                                                       GIRNSRT
C    UPDATE REGASP(IF NECESSARY)                                        GIRNSRT
   23 IF(IADD.EQ.REGASP)REGASP=LSTSPC(IADD)                             GIRNSRT
C                                                                       GIRNSRT
C   UPDATE AVAILABLE SPACE                                              GIRNSRT
      ISUB=NODSPC(IADD)
      LSTSPC(ISUB)=LSTSPC(IADD)
      ISUB=LSTSPC(IADD)
      NODSPC(ISUB)=NODSPC(IADD)
C                                                                       GIRNSRT
C    INSERT FUNCTION                                                    GIRNSRT
   24 NODSPC(IADD)=IARG                                                 GIRNSRT
      LSTSPC(IADD)=LSTTMP                                               GIRNSRT
      LNKSPC(IADD)=IADD                                                 GIRNSRT
      FLGSPC(IADD)=FLGSPC(IADD) .OR. FLGTMP
      FLGSPC(IADD)=FLGSPC(IADD) .OR. FL4MSK
      FLGSPC(IADD)=FLGSPC(IADD) .OR. FL5MSK
C                                                                       GIRNSRT
C   INSERT ANY ADDITIONAL FUNCTION VALUES                               GIRNSRT
      HEAD=IADD                                                         GIRNSRT
      OLDLOC=IADD                                                       GIRNSRT
      IF(NVAL.GT.1)GO TO 50                                             GIRNSRT
C                                                                       GIRNSRT
C   IF LAST CELL OF AVAILABLE SPACE WAS USED, WRITE MESSAGE             GIRNSRT
      IF(REGASP.EQ.LSTSPC(REGASP)) GO TO 909                            IZ03302
```

```
59•            IVAL=IABS(IVALS(I))                                          IZ11217
60•            RETURN                                                       GIRNSRT
61•      C                                                                  GIRNSRT
62•      C    IF THAT ADDRESS CONTAINS THE HEAD OF A CONFLICT LIST, GO TO 41 GIRNSRT
63•         25 ILOG=FL5MSK .AND. FLGSPC(IADD)
64•            IF(ILOG .GT. 0) GO TO 41
65•      C                                                                  GIRNSRT
66•      C    IF THAT ADDRESS CONTAINS A VALUE ON A MULTIVALUE LIST, GO TO 35 GIRNSRT
67•            ILOG=FL2MSK .AND. FLGSPC(IADD)
68•            ILOG2=FLOMSK .AND. FLGSPC(IADD)
69•            IF(ILOG .GT. 0 .AND. ILOG2 .EQ. 0) GO TO 35
70•      C                                                                  GIRNSRT
71•      C------------------------------------------------------------------ GIRNSRT
72•      C-THE ADDRESS CONTAINS A FUNCTION ON A CONFLICT LIST,BUT NOT THE HEAD OF GIRNSRT
73•            THIS=IADD                                                    GIRNSRT
74•      C                                                                  GIRNSRT
75•      C    FIND THE PRECEDING FUNCTION ON THE CONFLICT LIST              GIRNSRT
76•         26 ISUB=LNKSPC(THIS)
77•            IF(LNKSPC(ISUB) .EQ. IADD) GO TO 27
78•            THIS=LNKSPC(THIS)                                            GIRNSRT
79•            GO TO 26                                                     GIRNSRT
80•         27 LAST=LNKSPC(THIS)                                            GIRNSRT
81•            NEWLOC=REGASP                                                GIRNSRT
82•            IF(REGASP.EQ.LSTSPC(REGASP)) GO TO 98                        IZ03302
83•      C                                                                  GIRNSRT
84•      C    UPDATE AVAILABLE SPACE AND REGASP                             GIRNSRT
85•         28 ISUB=NODSPC(REGASP)
86•            LSTSPC(ISUB)=LSTSPC(REGASP)
87•            ISUB=LSTSPC(REGASP)
88•            NODSPC(ISUB)=NODSPC(REGASP)
89•            REGASP=LSTSPC(REGASP)                                        GIRNSRT
90•                                                                         GIRNSRT
91•      C    MOVE THE FUNCTION ON A CONFLICT LIST TO THE FIRST CELL OF AVAILABLE GIRNSRT
92•         29 NODSPC(NEWLOC)=NODSPC(IADD)                                  GIRNSRT
93•            LSTSPC(NEWLOC)=LSTSPC(IADD)                                  GIRNSRT
94•            LNKSPC(NEWLOC)=LNKSPC(IADD)                                  GIRNSRT
95•            FLGSPC(NEWLOC)=FLGSPC(IADD)                                  IZ08251
96•            FLGSPC(IADD)=0                                               IZ08251
97•            LNKSPC(LAST)=NEWLOC                                          GIRNSRT
98•      C                                                                  GIRNSRT
99•      C    INSERT THIS FUNCTION AS THE HEAD OF A CONFLICT LIST           GIRNSRT
100•           NODSPC(IADD)=IARG                                            GIRNSRT
101•           LNKSPC(IADD)=IADD                                            GIRNSRT
102•           LSTSPC(IADD)=LSTTMP                                          GIRNSRT
103•           FLGSPC(IADD)=(FLGTMP.OR.FLGSPC(IADD))                        GIRNSRT
104•           FLGSPC(IADD)=(FL5MSK.OR.FLGSPC(IADD))                        GIRNSRT
105•           FLGSPC(IADD)=FL4MSK .OR. FLGSPC(IADD)
106•           ILOG=FLGSPC(NEWLOC) .AND. FLOMSK
107•           IF(ILOG .EQ. 0) GO TO 34
108•     C                                                                  GIRNSRT
109•     C    IF THE FUNCTION THAT W.S MOVED IS THE HEAD OF A MULTIVALUE LIST,FIX GIRNSRT
110•           NEXT=LSTSPC(NEWLOC)                                          GIRNSRT
111•        30 NEXT=LSTSPC(NEXT)                                            GIRNSRT
112•           IF(LSTSPC(NEXT).NE.IADD)GO TO 30                             GIRNSRT
113•           LSTSPC(NEXT)=NEWLOC                                          GIRNSRT
114•     C                                                                  GIRNSRT
115•     C    INSERT ANY ADDITIONAL FUNCTION VALUES                         GIRNSRT
116•        34 HEAD=IADD                                                    GIRNSRT
```

```
      OLDLOC=IADD                                                       GIRNSRT
      IF(NVAL.GT.1)GO TO 50                                             GIRNSRT
      IF(REGASP.EQ.LSTSPC(REGASP)) GO TO 909                            IZ03302
      IVAL=IABS(IVALS(I))                                               IZ11217
      RETURN                                                            GIRNSRT
C                                                                       GIRNSRT
C-----------------------------------------------------------------------GIRNSRT
C-THE ADDRESS CONTAINS A VALUE ON A MULTIVALUE LIST                     GIRNSRT
   35 NEWLOC=REGASP                                                     GIRNSRT
      IF(REGASP.EQ.LSTSPC(REGASP)) GO TO 98                             IZ03302
C                                                                       GIRNSRT
C   UPDATE AVAILABLE SPACE AND REGASP                                   GIRNSRT
   36 ISUB=NODSPC(REGASP)
      LSTSPC(ISUB)=LSTSPC(REGASP)
      ISUB=LSTSPC(REGASP)
      NODSPC(ISUB)=NODSPC(REGASP)
      REGASP=LSTSPC(REGASP)                                             GIRNSRT
C                                                                       GIRNSRT
C   MOVE THE VALUE ON A MULTIVALUE LIST TO THE FIRST CELL OF AVAILABLE S GIRNSRT
   37 NODSPC(NEWLOC)=NODSPC(IADD)                                       GIRNSRT
      LSTSPC(NEWLOC)=LSTSPC(IADD)                                       GIRNSRT
      LNKSPC(NEWLOC)=LNKSPC(IADD)                                       GIRNSRT
      FLGSPC(NEWLOC)=FLGSPC(IADD)                                       IZ08251
      FLGSPC(IADD)=0                                                    IZ08251
C                                                                       IZ03272
C     RESET POINTERS                                                    IZ03272
C                                                                       IZ03272
      LI=LSTSPC(NEWLOC)                                                 IZ03272
      ILOG=FLOMSK .AND. FLGSPC(LI)
      IF(ILOG .EQ. 0) GO TO 200
      ISUB=LSTSPC(LI)
      LNKSPC(ISUB)=NEWLOC
      GO TO 201                                                         IZ03272
  200 LNKSPC(LI)=NEWLOC                                                 IZ03272
  201 ISUB=LNKSPC(NEWLOC)
      KZVAL=LSTSPC(ISUB)
      ILOG=FLGSPC(KZVAL) .AND. FLOMSK
      IF(ILOG .NE. 0) GO TO 38
      ISUB=LNKSPC(NEWLOC)
      LSTSPC(ISUB)=NEWLOC
      GO TO 39                                                          IZ11301
   38 LSTSPC(KZVAL)=NEWLOC                                              IZ11301
   39 NODSPC(IADD)=IARG                                                 IZ11301
C  INSERT THIS FUNCTION AS THE HEAD OF A CONFLICT LIST                  GIRNSRT
      LNKSPC(IADD)=IADD                                                 GIRNSRT
      LSTSPC(IADD)=LSTTMP                                               GIRNSRT
      FLGSPC(IADD)=(FLGTMP.OR.FLGSPC(IADD))                             GIRNSRT
      FLGSPC(IADD)=(FL5MSK.OR.FLGSPC(IADD))                             GIRNSRT
      FLGSPC(IADD)=FL4MSK .OR. FLGSPC(IADD)
      IF(REGASP.EQ.LSTSPC(REGASP)) GO TO 909                            IZ03302
      IVAL=IABS(IVALS(I))                                               IZ11217
      RETURN                                                            GIRNSRT
C                                                                       GIRNSRT
C-----------------------------------------------------------------------GIRNSRT
C-THE ADDRESS CONTAINS THE HEAD OF A CONFLICT LIST                      GIRNSRT
   41 THIS=IADD                                                         GIRNSRT
C                                                                       GIRNSRT
C   IF THE FUNCTION TO BE INSERTED IS NOT ON THE CONFLICT LIST, GO TO 60 GIRNSRT
```

```
      42  IF(NODSPC(THIS).EQ.IARG)GO TO 43                              GIRNSRT
          ISUB=LNKSPC(THIS)
          ILOG=FLGSPC(ISUB) .AND. FLSMSK
          IF(ILOG .NE. 0) GO TO 60
          THIS=LNKSPC(THIS)                                             GIRNSRT
          GO TO 42                                                      GIRNSRT
C                                                                       GIRNSRT
C-------------------------------------------------------------------- GIRNSRT
C-THE FUNCTION TO BE INSERTED IS ON THE CONFLICT LIST                   GIRNSRT
      43  HEAD=THIS                                                     GIRNSRT
          ILOG=FLOMSK .AND. FLGSPC(THIS)
          IF(ILOG .EQ. 0) GO TO 51
          NEXT=LSTSPC(THIS)                                             IZ06097
C                                                                       IZ04197
C     OLDLOC IS THE LOCATION OF THE LAST VALUE ON THE MULTIVALUE LIST   IZ04197
C                                                                       IZ04197
          OLDLOC=LNKSPC(NEXT)                                           IZ04197
C                                                                       GIRNSRT
C-------------------------------------------------------------------- GIRNSRT
C-INSERT ADDITIONAL FUNCTION VALUES                                     GIRNSRT
      50  LSTASP=NODSPC(REGASP)                                         GIRNSRT
          IN=0                                                          GIRNSRT
          GO TO 56                                                      GIRNSRT
C                                                                       GIRNSRT
C-------------------------------------------------------------------- GIRNSRT
C-FORM MULTIVALUE LIST TO ADD VALUE(S) TO SINGLE-VALUED FUNCTION        GIRNSRT
      51  IN=0                                                          GIRNSRT
          IF(REGASP.EQ.LSTSPC(REGASP))GO TO 98                          GIRNSRT
          LSTASP=NODSPC(REGASP)                                         GIRNSRT
          NEWLOC=REGASP                                                 GIRNSRT
          REGASP=LSTSPC(REGASP)                                         GIRNSRT
          NODSPC(NEWLOC)=LSTSPC(THIS)                                   IZ06097
          TEMP=(FLGSPC(THIS).AND.FLG67)                                 IZ07021
          FLGSPC(NEWLOC)=(TEMP.OR.FLGSPC(NEWLOC))                       GIRNSRT
          FLGSPC(THIS)=(FLGSPC(THIS).AND.NFLG67)                        IZ07021
          FLGSPC(THIS)=(FL2MSK.OR.FLGSPC(THIS))                         GIRNSRT
          FLGSPC(THIS)=(FLOMSK.OR.FLGSPC(THIS))                         GIRNSRT
          OLDLOC=THIS                                                   GIRNSRT
C                                                                       GIRNSRT
C-------------------------------------------------------------------- GIRNSRT
C   INSERT ANOTHER VALUE ON MULTIVALUE LIST                             GIRNSRT
      52  FLGSPC(NEWLOC)=(FL2MSK.OR.FLGSPC(NEWLOC))                    GIRNSRT
          FLGSPC(NEWLOC)=(FLIMSK.OR.FLGSPC(NEWLOC))                     GIRNSRT
          LSTSPC(OLDLOC)=NEWLOC                                         GIRNSRT
          LNKSPC(NEWLOC)=OLDLOC                                         GIRNSRT
      55  OLDLOC=NEWLOC                                                GIRNSRT
      56  NEWLOC=REGASP                                                 GIRNSRT
          IF(IN.GT.0)GO TO 57                                           GIRNSRT
C                                                                       GIRNSRT
C   NO VALUES HAVE BEEN INSERTED YET                                    GIRNSRT
          IN=1                                                          GIRNSRT
          GO TO 58                                                      GIRNSRT
C                                                                       GIRNSRT
C   SOME VALUES HAVE BEEN INSERTED                                      GIRNSRT
      57  IF(IN.EQ.NVAL)GO TO 67                                        GIRNSRT
          IN=IN+1                                                       GIRNSRT
C                                                                       GIRNSRT
      58 IF(REGASP.EQ.LSTSPC(REGASP)) GO TO 909                         IZ11277
```

```
233•       581 REGASP=LSTSPC(REGASP)                                    GIRNSRT
234•       582 NODSPC(NEWLOC)=IVALS(IN)                                 GIRNSRT
235•           FLGSPC(NEWLOC)=(ITYP(IN).OR.FLGSPC(NEWLOC))              GIRNSRT
236•           GO TO 52                                                 GIRNSRT
237•     C                                                              GIRNSRT
238•     C   END MULTIVALUE LIST AND UPDATE AVAILABLE SPACE             GIRNSRT
239•        67 LSTSPC(OLDLOC)=HEAD                                      GIRNSRT
240•           NODSPC(REGASP)=LSTASP                                    GIRNSRT
241•           LSTSPC(LSTASP)=REGASP                                    GIRNSRT
242•           IVAL=IABS(IVALS(I))                                      IZ11217
243•           ISUB=LSTSPC(HEAD)
244•           LNKSPC(ISUB)=OLDLOC
245•           NVAL=IN                                                  GIRNSRT
246•           IF(REGASP.EQ.LSTSPC(REGASP)) GO TO 909                   IZ11277
247•           RETURN                                                   GIRNSRT
248•     C                                                              GIRNSRT
249•     C---------------------------------------------------------------GIRNSRT
250•     C-THE FUNCTION TO BE INSERTED IS NOT ON THE CONFLICT LIST      GIRNSRT
251•        60 ASPREG=REGASP                                            GIRNSRT
252•           LSTASP=NODSPC(REGASP)                                    GIRNSRT
253•           IF(REGASP.EQ.LSTSPC(REGASP)) GO TO 98                    IZ03302
254•     C                                                              GIRNSRT
255•     C   UPDATE AVAILABLE SPACE AND REGASP                          GIRNSRT
256•       601 ISUB=NODSPC(REGASP)
257•           LSTSPC(ISUB)=LSTSPC(REGASP)
258•           ISUB=LSTSPC(REGASP)
259•           NODSPC(ISUB)=NODSPC(REGASP)
260•           REGASP=LSTSPC(REGASP)                                    GIRNSRT
261•     C                                                              GIRNSRT
262•     C   INSERT FUNCTION IN FIRST CELL OF AVAILABLE SPACE           GIRNSRT
263•        61 NODSPC(ASPREG)=IARG                                      GIRNSRT
264•           IF(NVAL.EQ.1)GO TO 611                                   GIRNSRT
265•           LSTSPC(ASPREG)=REGASP                                    GIRNSRT
266•           FLGSPC(ASPREG)=(FL2MSK.OR.FLGSPC(ASPREG))                GIRNSRT
267•           FLGSPC(ASPREG)=(FL0MSK.OR.FLGSPC(ASPREG))                GIRNSRT
268•           GO TO 612                                                GIRNSRT
269•       611 LSTSPC(ASPREG)=IVALS(I)                                  GIRNSRT
270•       612 FLGSPC(ASPREG)=(ITYP(I).OR.FLGSPC(ASPREG))               GIRNSRT
271•           FLGSPC(ASPREG)=(FL1MSK.OR.FLGSPC(ASPREG))                GIRNSRT
272•           FLGSPC(ASPREG)=FL4MSK .OR. FLGSPC(ASPREG)
273•           LNKSPC(ASPREG)=LNKSPC(THIS)                              GIRNSRT
274•           LNKSPC(THIS)=ASPREG                                      GIRNSRT
275•           IF(NVAL.EQ.1)GO TO 613                                   GIRNSRT
276•     C                                                              GIRNSRT
277•     C   INSERT ADDITIONAL VALUES                                   GIRNSRT
278•           LSTASP=NODSPC(REGASP)                                    GIRNSRT
279•           OLDLOC=ASPREG                                            GIRNSRT
280•           HEAD=ASPREG                                              GIRNSRT
281•           IN=0                                                     GIRNSRT
282•           GO TO 56                                                 GIRNSRT
283•       613 IF(REGASP.EQ.LSTSPC(REGASP)) GO TO 909                   IZ03302
284•           IVAL=IABS(IVALS(I))                                      IZ11217
285•           RETURN                                                   GIRNSRT
286•     C                                                              IZ04142
287•     C     DESTRUCTIVE INSERTION                                    IZ04142
288•     C                                                              IZ04142
289•       350 IADDI=IADD                                               IZ04142
290•           INDEX=0                                                  IZ04142
```

```
2910          CALL LVFIND(INDEX,INDEX,INDEX,INDEX)                    IZ05177
2920          FLGSPC(IADD)=FLGSPC(IADD).OR.FL4MSK                     COMPAND
2930          IF(IVAL.EQ.-1) GO TO 90                                 IZ04142
2940          IF(LSTHED) 354,90,356                                   IZ04142
2950      354 LSTSPC(IADD)=IVALS(I)                                   IZ04142
2960          GO TO 365                                               IZ04142
2970      356 NODSPC(IADD)=IVALS(I)                                   IZ04142
2980      365 FLGSPC(IADD)=FLGSPC(IADD).AND.NFLG67                    IZ04142
2990          FLGSPC(IADD)=FLGSPC(IADD).OR.ITYP(I)                    IZ04142
3000          GO TO 360                                               IZ04142
3010       90 IF(IPOS)91,99,92                                        IZ04142
3020       91 IPOS=IPOS+1                                             IZ06287
3030          GO TO 93                                                IZ06287
3040       92 IPOS=IPOS-1                                             IZ06287
3050       93 IADD=IADD1                                              IZ06287
3060          IF(IPOS.EQ.0) GO TO 125                                 IZ06287
3070          INDEX=0                                                 IZ04142
3080          CALL LVFIND(INDEX,INDEX,INDEX,INDEX)                    IZ05177
3090          IF(IVAL.EQ.-1) GO TO 99                                 IZ04142
3100          IF(IPOS .LT. 0) GO TO 370
3110          IADD=IADD1                                              IZ04142
3120          GO TO 125                                               IZ04142
3130      370 NEWLOC=REGASP                                           IZ04142
3140          IF(LSTHED) 325,99,375                                   IZ04142
3150      C   UPDATE AVAILABLE SPACE                                  IZ04142
3160      375 ISUB=NODSPC(REGASP)
3170          LSTSPC(ISUB)=LSTSPC(REGASP)
3180          ISUB=LSTSPC(REGASP)
3190          NODSPC(ISUB)=NODSPC(REGASP)
3200          REGASP=LSTSPC(REGASP)                                   IZ04142
3210          GO TO 377                                               IZ04142
3220      C                                                           IZ04142
3230      C   NONDESTRUCTIVE INSERTION                                IZ04142
3240      C                                                           IZ04142
3250      300 IADD1=IADD                                              IZ04142
3260          NEWLOC=REGASP                                           IZ04142
3270          INDEX=0                                                 IZ04142
3280          CALL LVFIND(INDEX,INDEX,INDEX,INDEX)                    IZ05177
3290          FLGSPC(IADD)=FLGSPC(IADD).OR.FL4MSK                     COMPAND
3300          IF(IVAL.EQ.-1) GO TO 90                                 IZ04142
3310          IF(LSTHED) 344,90,346                                   IZ04142
3320      344 IF(IPOS .GT. 0) GO TO 325
3330          IADD=IADD1                                              IZ04142
3340          GO TO 125                                               IZ04142
3350      C   CREATE MULTIVALUE LIST                                  IZ04142
3360      325 ISUB=NODSPC(REGASP)
3370          LSTSPC(ISUB)=LSTSPC(REGASP)
3380          ISUB=LSTSPC(REGASP)
3390          NODSPC(ISUB)=NODSPC(REGASP)
3400          REGASP=LSTSPC(REGASP)                                   IZ04142
3410          IF(REGASP.EQ.LSTSPC(REGASP)) GO TO 909                  IZ04142
3420          NWLOC2=REGASP                                           IZ05177
3430      C   UPDATE AVAILABLE SPACE                                  IZ04142
3440          ISUB=NODSPC(REGASP)
3450          LSTSPC(ISUB)=LSTSPC(REGASP)
3460          ISUB=LSTSPC(REGASP)
3470          NODSPC(ISUB)=NODSPC(REGASP)
3480          REGASP=LSTSPC(REGASP)                                   IZ04142
```

```
                NODSPC(NEWLOC)=IVALS(1)                                 IZ04172
                LSTSPC(NEWLOC)=NWLOC2                                   IZ05177
                LNKSPC(NEWLOC)=NWLOC2                                   IZ05177
                FLGSPC(NEWLOC)=FLGTMP.OR.FL2MSK                         IZ05177
                NODSPC(NWLOC2)=LSTSPC(IADD)                             IZ05177
                LSTSPC(NWLOC2)=IADD                                     IZ05177
                LNKSPC(NWLOC2)=NEWLOC                                   IZ05177
                KLGTEP=FLGSPC(IADD).AND.FLG67                           IZ05177
                FLGSPC(NWLOC2)=(FLIMSK.OR.FL2MSK).OR.KLGTEP             IZ05177
                LSTSPC(IADD)=NEWLOC                                     IZ04172
                FLGSPC(IADD)=(FLGSPC(IADD).OR.FLOMSK).OR.FL2MSK         IZ04172
          320   IF(REGASP.EQ.LSTSPC(REGASP)) GO TO 909                  IZ04142
          360   IVAL=IABS(IVALS(1))                                     IZ11217
                RETURN                                                  IZ04142
      C         UPDATE AVAILABLE SPACE                                  IZ04142
          346   ISUB=NODSPC(REGASP)
                LSTSPC(ISUB)=LSTSPC(REGASP)
                ISUB=LSTSPC(REGASP)
                NODSPC(ISUB)=NODSPC(REGASP)
                REGASP=LSTSPC(REGASP)                                   IZ04142
                IF(IPOS .LT. 0) GO TO 347
          377   ISTLOC=LNKSPC(IADD)                                     IZ04142
                NODSPC(NEWLOC)=IVALS(1)                                 IZ04142
                LSTSPC(NEWLOC)=IADD                                     IZ04142
                LNKSPC(NEWLOC)=ISTLOC                                   IZ04142
                FLGSPC(NEWLOC)=FLGTMP.OR.FL2MSK                         IZ04142
                ISUB=LSTSPC(ISTLOC)
                ILOG=FLGSPC(ISUB) .AND. FLOMSK
                IF(ILOG .EQ. 0) GO TO 321
                ISUB=LSTSPC(ISTLOC)
                LSTSPC(ISUB)=NEWLOC
                GO TO 322                                               IZ04142
          321   LSTSPC(ISTLOC)=NEWLOC                                   IZ04142
          322   LNKSPC(IADD)=NEWLOC                                     IZ04142
                GO TO 320                                               IZ04142
          347   NODSPC(NEWLOC)=IVALS(1)                                 IZ04142
                LSTSPC(NEWLOC)=LSTSPC(IADD)                             IZ04142
                LNKSPC(NEWLOC)=IADD                                     IZ04142
                FLGSPC(NEWLOC)=FLGTMP.OR.FL2MSK                         IZ04142
                ISUB=LSTSPC(IADD)
                ILOG=FLGSPC(ISUB) .AND. FLOMSK
                IF(ILOG .EQ. 0) GO TO 323
                KZVAL=LSTSPC(IADD)                                      IZ04142
                ISUB=LSTSPC(KZVAL)
                LNKSPC(ISUB)=NEWLOC
                GO TO 324                                               IZ04142
          323   ISUB=LSTSPC(IADD)
                LNKSPC(ISUB)=NEWLOC
          324   LSTSPC(IADD)=NEWLOC                                     IZ04142
                GO TO 320                                               IZ04142
           98   IVAL=-3                                                 IZ06287
                PRINT 20001                                             IZ03302
        20001   FORMAT(1X,78H ERROR...THERE IS NO ADDITIONAL SPACE FOR THE GRAPH,  RR1
               *THE PROGRAM IS TERMINATED)                              RR1
                STOP                                                    GIRNSRT
           99   IVAL=-1                                                 IZ06287
            2   FORMAT(6H ONLY ,I4,28H VALUE(S) HAVE BEEN INSERTED)     GIRNSRT
           22   FORMAT(1X,I5,1H(,I5,35H) USED LAST CELL OF AVAILABLE SPACE)  GIRNSRT
                RETURN                                                  GIRNSRT
          909   PRINT 22,IFUNC,IARG                                     IZ03302
      C     THIS INSERTION HAS FILLED GIRS MEMORY - CALL A USER SUPPLIED IZ11137
      C     PROGRAM - LVEXIT.                                           IZ11137
                IVAL=-1                                                 IZ11137
                RETURN                                                  IZ08121
                END                                                     GIRNSRT
```

```
1•            SUBROUTINE LVSETP                                                IZ05177
2•            INTEGER FLGSPC,FLAGSP,REGASP,BINFIL,FLOMSK,FL1MSK,FL2MSK,FL5MSK,  IZ05177
3•           * FL3MSK,FL4MSK,FLG67,SEQSPC                                      IZ05177
4•            COMMON/LVFLAG/FLOMSK,FL1MSK,FL2MSK,FL3MSK,FL4MSK,FL5MSK,FLG67    IZ05177
5•            COMMON/LVVTR5/BINFIL,KOMPAN,NODESP(1)/LVVTR6/LISTSP(1)           IZ05177
6•           * /LVVTR7/LINKSP(1)/LVVTR8/FLAGSP(1)                              IZ05177
7•            COMMON /LVTABL/ MAPSZE,MAP(1) /LVVSEQ/ ISEQSZ,SEQSPC(1)          IZ05177
8•            COMMON/LVVTR1/MEMSZE,REGASP,NODSPC(    1)/LVVTR2/LSTSPC(    1)/  GIRSETP
9•           *LVVTR3/LNKSPC(    1)/LVVTR4/FLGSPC(    1)                        GIRSETP
10•           COMMON/LVRAND/KPRIME,KSEED,NROW,KDNODE,KDROW,KTEMP
11•           DATA FLOMSK/128/,FL1MSK/64/,FL2MSK/32/,FL5MSK/4/,FLG67/3/,       RR1
12•          *FL3MSK/16/,FL4MSK/8/                                             RR1
13•           KSEED=KPRIME/2
14•           NROW=KSEED
15•           KTEMP=KSEED-KPRIME
16•           KDNODE=KPRIME
17•           REGASP=1                                                         GIRSETP
18•           DO 10 I=2,MEMSZE                                                 GIRSETP
19•           LNKSPC(I)=0                                                      GIRSETP
20•           FLGSPC(I)=0                                                      GIRSETP
21•           NODSPC(I)=I-1                                                    GIRSETP
22•     10    LSTSPC(I-1)=I                                                    GIRSETP
23•           FLGSPC(1)=0                                                      GIRSETP
24•           LNKSPC(1)=0                                                      GIRSETP
25•           NODSPC(1)=MEMSZE                                                 GIRSETP
26•           LSTSPC(MEMSZE)=1                                                 GIRSETP
27•           RETURN                                                           GIRSETP
28•           END                                                              GIRSETP
```

```
      SUBROUTINE MODID(MODE)
      DIMENSION IBUF(80),IEND(3)
      IF(MODE .NE. 0) GO TO 5
      WRITE(13,1)
    1 FORMAT(5X,16H OUTPUT DEVICE X)
      WRITE(14,2)
    2 FORMAT(5X,16H OUTPUT DEVICE Y)
      WRITE(15,3)
    3 FORMAT(5X,16H OUTPUT DEVICE Z)
      WRITE(6,6)
    6 FORMAT(1H1,52X,27H RESULTS OF ROLL CALL CHECK)
    5 DO 10 I=10,12
      ENDFILE I
      REWIND I
      IEND(I-9)=0
      READ(I,7,END=8) ICHAR
    7 FORMAT(A1)
      GO TO 10
    8 IEND(I-9)=1
   10 CONTINUE
      IEOF=0
      DO 15 I=1,3
      IF(IEND(I) .NE. 0) GO TO 15
      IF(IEOF .EQ. 1) GO TO 40
      IEOF=1
      IOUT=9+I
   15 CONTINUE
      IF(IEOF .EQ. 0) GO TO 50
      REWIND IOUT
      IOUT2=IOUT+3
      WRITE(IOUT2,20) MODE
   20 FORMAT(//20X,12H MODE INDEX=,I3)
      DO 30 I=1,100
      READ(IOUT,25,END=60) (IBUF(J),J=1,80)
   25 FORMAT(80A1)
   30 WRITE(IOUT2,25) (IBUF(J),J=1,80)
   40 WRITE(6,45) MODE
   45 FORMAT(//21X,86H **ERROR IN ROLL CALL CHECK - MORE THAN 1 OUTPUT D
     *EVICE WAS WRITTEN ON FOR MODE INDEX ,I3,2H**)
      GO TO 60
   50 WRITE(6,55) MODE
   55 FORMAT(//25X,79H **ERROR IN ROLL CALL CHECK - NO OUTPUT DEVICES WE
     *RE WRITTEN ON FOR MODE INDEX ,I3,2H**)
   60 ENDFILE 10
      ENDFILE 11
      ENDFILE 12
      REWIND 10
      REWIND 11
      REWIND 12
      RETURN
      END
```

```
      FUNCTION NEXT(IA)                                                 NEXT   2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     • JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      INTEGER A,BLANK                                                   NEXT   4
      DATA BLANK/1H /                                                   NEXT   5
      IF(IA .GT. N) GO TO 15                                            NEXT   6
      DO 10 I=IA,N                                                      NEXT   7
      IF(A(I) .EQ. BLANK) GO TO 10                                      NEXT   8
      NEXT=A(I)                                                         NEXT   9
      JPTR=I+1                                                          NEXT  10
      RETURN                                                            NEXT  11
   10 CONTINUE                                                          NEXT  12
      NEXT=BLANK
      JPTR=N+1                                                          NEXT  14
      RETURN                                                            NEXT  15
   15 NEXT=BLANK
      JPTR=IA+1
      RETURN
      END                                                               NEXT  16
```

```
      FUNCTION NXTBLK(ILOC,IEND)                                        NXTBLK 2
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      COMMON/LABELS/STATRA(2,200),NLABEL                                NXTBLK 4
      INTEGER STATRA,BITGET                                             NXTBLK 5
      I=IBLOCK(ILOC)                                                    NXTBLK 6
      IF(I .EQ. 999) GO TO 10                                           NXTBLK 7
      IF(I .EQ. 998) GO TO 5                                            NXTBLK 8
      NXTBLK=BITGET(STATRA(2,I),36,18)                                  NXTBLK 9
      RETURN                                                            NXTBLK10
    5 NXTBLK=IEND+1                                                     NXTBLK11
      IF(NXTBLK .GT. NBLOCK) CALL ERROR(38,KDM1,KDM2)
      RETURN                                                            NXTBLK13
   10 NXTBLK=0                                                          NXTBLK14
      RETURN                                                            NXTBLK15
      END                                                               NXTBLK16
```

```
      SUBROUTINE PARSE
      COMMON/LVARGS/LVFUNC,LVVARG,LVVAD,LVVPOS,LVVTYP,LVVAL,
     +LVHEAD,LVVNVL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP
      COMMON/LVTABL/LVTSIZ,LVMAP(     1)/LVVSEQ/LVSIZE,LVSQSP(     1)
      COMMON/NEEDS/STJ,JSTACK,R,JAS,J,JLAST,RTEMP,STACK(400,4)
      COMMON/FUNC/NARY(5,17),MARGS,IARGS(50),FNCLOC(5),NFUNC
      COMMON /STRING/ NTYPE,NSTR,STR
      COMMON /GIRL/NTERMS,PLUS,MINUS,SLASH,LPAR,RPAR,COMMA,STAR,EXP,LT,
     +LE,GT,GE,EQ,NE,OR,AND,NOT,EQUALS,OPRAND
      COMMON /HL/ HOL,ACTION,FUNC1,FUNC2,FUNC3,LEFT,RIGHT,STRING,MAXJ
      COMMON /NEED/ START,ASSOC,LEVEL,STOP
      COMMON /TYP/ NARRAY,TYPE1,TYPE2,ERRFLG
      COMMON/NOPAR/NOPAR,NDEP,NDEPTH,NFLAG
      COMMON /NTIMES/ NTIMES,I
      COMMON/VAR/VFOR(15),NUMCHR,NCHRP,CHAR,NDICT
      INTEGER TYPE1,TYPE2,START,TYP(3)
      LOGICAL ERRFLG,FAIL
      INTEGER STR(1),STEMP,ST,DICT(19)
      EQUIVALENCE(DICT(1),PLUS)
      INTEGER PLUS,MINUS,SLASH,LPAR,RPAR,COMMA,STAR,EXP,LT,LE,GT,GE,EQ,
     +NE,OR,AND,NOT,EQUALS,OPRAND,ASSOC,LEVEL,STOP,ACTION,HOL,LEFT,RIGHT
     +,STRING,FUNC1,FUNC2,FUNC3
      DATA NTIMES /0/
      IF(NTIMES .GT. 0) GO TO 3
      NTIMES=1
C     EXECUTE
      GO TO 25000
25001 CONTINUE
      CALL PHONEY
      CALL LVFECH(19)
      READ(19)PLUS,MINUS,SLASH,LPAR,RPAR,COMMA,STAR,EXP,LT,LE,GT,GE,EQ,
     +NE,OR,AND,NOT,EQUALS,OPRAND,ASSOC,LEVEL,STOP,ACTION,HOL,LEFT,RIGHT
     +,STRING,FUNC1,FUNC2,FUNC3,NTERMS,(TYP(I),I=1,3)
    3 IF(NSTR .LE. 0) RETURN
      ERRFLG=.FALSE.
      START=TYP(NTYPE)
      NARGS=0
      MAXJ=0
      NOPAR=0
      TYPE1=-1
      TYPE2=-1
      NARRAY=-1
```

```
      NDEPTH=0
      NDEP=0
      NFLAG=0
      DO 20 I=1,50
   20 IARGS(I)=0
      DO 22 I=1,85
   22 NARY(I)=0
      DO 10 I=1,NSTR
C     STRING(+HOL.I @NTEMP//4,HOL $ @NTEMP)
      LVI    AAB =      STRING
C**** LVI    AAB              +          HOL
      LVVPOS =            I
      LVVTYP =   3
      LVFUNC=         HOL
      LVVARG= LVI    AAB
      CALL LVFIND(LV2       A,LV2       B,LV2       C,LV2       D)
      LVI    AAC = LVI    AAB
      IF (LVVAL.NE.-1) LVI    AAC = LVVAL
C**** LVI    AAC              @         NTEMP
           NTEMP = LVI    AAC
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO            4
      CALL LVGRN(LVI    AAC)
C**** LVI    AAB           HOL  LVI    AAC
      LVDEST=  0
      LVTYPE(1) =  0
      LVVALS(1) = LVI    AAC
      LVVNVL =  1
      LVFUNC =         HOL
      LVVARG=LVI    AAB
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
C**** LVI    AAC              @         NTEMP
           NTEMP = LVI    AAC
    4 CONTINUE
      IF(ERRFLG) GO TO 25
      ST=IABS(STR(I))
      IF(STR(I) .LT. 0) GO TO 6
C     NTEMP HOL @@ST@@
C****      NTEMP           HOL           @@
      LVDEST=  0
      LVI    AAB = ST
      LVTYPE(1) =  1
      LVVALS(1) = LVI    AAB
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =         HOL
      LVVARG=     NTEMP
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      IF(ERRFLG) GO TO 25
C     STRING STRING OPRAND//10
C****      STRING       STRING       OPRAND
      LVDEST=  0
      LVTYPE(1) =  0
```

```
101*            LVVALS(1) =      OPRAND
102*            LVVNVL =  1
103*            LVFUNC =      STRING
104*            LVVARG=     STRING
105*            CALL LVNSRT
106*            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
107*            IF(LVVAL.LT.0) RETURN
108*            LVVTR = LVVAL
109*            LVVAL = -100
110*            IF (LVVTR.NE.-1) GO TO          10
111*            IF(ERRFLG) GO TO 25
112*           & STEMP=DICT(ST)
113*      C     STRING STRING @@STEMP@@
114*      C****      STRING       STRING           @@
115*            LVDEST=  0
116*            LV1     AAC = STEMP
117*            LVTYPE(1) =  1
118*            LVVALS(1) = LV1     AAC
119*            LVDEST=  0
120*            LVVNVL =  1
121*            LVFUNC =      STRING
122*            LVVARG=     STRING
123*            CALL LVNSRT
124*            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
125*            IF(LVVAL.LT.0) RETURN
126*            IF(ERRFLG) GO TO 25
127*         10 CONTINUE
128*            LV1     AAD =      OPRAND
129*            LVDEST=  0
130*            LV1     AAE = 0
131*            LVTYPE(1) =  1
132*            LVVALS(1) = LV1     AAE
133*            LVDEST=  0
134*            LVVNVL =  1
135*            LVFUNC =       FUNC2
136*            LVVARG=LV1     AAD
137*            CALL LVNSRT
138*            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
139*            IF(LVVAL.LT.0) RETURN
140*            LVDEST=  0
141*            LV1     AAF = 0
142*            LVTYPE(1) =  1
143*            LVVALS(1) = LV1     AAF
144*            LVDEST=  0
145*            LVVNVL =  1
146*            LVFUNC =       FUNC3
147*            LVVARG=LV1     AAD
148*            CALL LVNSRT
149*            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
150*            IF(LVVAL.LT.0) RETURN
151*            LVDEST=  0
152*            LV1     AAG = 0
153*            LVTYPE(1) =  1
154*            LVVALS(1) = LV1     AAG
155*            LVDEST=  0
156*            LVVNVL =  1
157*            LVFUNC =       LEVEL
158*            LVVARG=LV1     AAD
```

```
159•            CALL LVNSRT
160•            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
161•            IF(LVVAL.LT.0) RETURN
162•            CALL RECOG(FAIL)
163•            IF(FAIL) GO TO 40
164•         25 CONTINUE
165•            IF(ERRFLG) PRINT 100
166•        100 FORMAT(/)
167•            CALL PRNTS
168•            NCHAR=0
169•         30 NCHAR=NCHAR+1
170•      C     STRING+HOL,NCHAR/35(-LEFT,-RIGHT,-HOL,-STRING/30/30)
171•      C++++      STRING              +          HOL
172•            LVVPOS =       NCHAR
173•            LVVTYP =    3
174•            LVFUNC=         HOL
175•            LVVARG=      STRING
176•            CALL LVFIND(LV2       E,LV2       F,LV2       G,LV2       H)
177•            LV1     AAD =      STRING
178•            IF (LVVAL.NE.-1) LV1     AAD = LVVAL
179•            LVVTR = LVVAL
180•            LVVAL = -100
181•            IF (LVVTR.EQ.-1) GO TO           35
182•            LV1     AAH = LV1     AAD
183•      C++++ LV1     AAE             -          LEFT
184•            LVVAD=-1
185•            LVVTYP=-1
186•            LVVPOS=1
187•            LVFUNC=       LEFT
188•            LVVARG=LV1     AAH
189•            CALL LVDLET
190•            LV1     AAH = LV1     AAD
191•      C++++ LV1     AAE             -         RIGHT
192•            LVVAD=-1
193•            LVVTYP=-1
194•            LVVPOS=1
195•            LVFUNC=      RIGHT
196•            LVVARG=LV1     AAH
197•            CALL LVDLET
198•            LV1     AAH = LV1     AAD
199•      C++++ LV1     AAE             -          HOL
200•            LVVAD=-1
201•            LVVTYP=-1
202•            LVVPOS=1
203•            LVFUNC=         HOL
204•            LVVARG=LV1     AAH
205•            CALL LVDLET
206•            LV1     AAH = LV1     AAD
207•      C++++ LV1     AAE             -       STRING
208•            LVVAD=-1
209•            LVVTYP=-1
210•            LVVPOS=1
211•            LVFUNC=     STRING
212•            LVVARG=LV1     AAH
213•            CALL LVDLET
214•            LVVTR = LVVAL
215•            LVVAL = -100
216•            IF (LVVTR.EQ.-1) GO TO           30
```

```
      IF (LVVTR.NE.-1) GO TO          30
   35 CONTINUE
C     STRING-STRING
C****      STRING              -       STRING
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=     STRING
      LVVARG=     STRING
      CALL LVDLET
C     OPRAND(-OPRAND,-STRING,-ACTION,-FUNC1)
      LVI     AAD =      OPRAND
C**** LVI     AAD                -       OPRAND
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=     OPRAND
      LVVARG=LVI     AAD
      CALL LVDLET
C**** LVI     AAD                =       STRING
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=     STRING
      LVVARG=LVI     AAD
      CALL LVDLET
C**** LVI     AAD                -       ACTION
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=     ACTION
      LVVARG=LVI     AAD
      CALL LVDLET
C**** LVI     AAD                -       FUNC1
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=     FUNC1
      LVVARG=LVI     AAD
      CALL LVDLET
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=     FUNC2
      LVVARG=LVI     AAD
      CALL LVDLET
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=     FUNC3
      LVVARG=LVI     AAD
      CALL LVDLET
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=     LEVEL
      LVVARG=LVI     AAD
      CALL LVDLET
```

```
      NSTR=NCHRP
      RETURN
   90 PRINT 300,MAXJ
  300 FORMAT(1X,34H PARSE FAILED AFTER CHARACTER NO. ,I3)
      ERRFLG=.TRUE.
      GO TO 25
C     COMPLETE
25000 CONTINUE
      LV2A=0
      LV2B=0
      LV2C=0
      LV2D=0
      LV2E=0
      LV2F=0
      LV2G=0
      LV2H=0
      GO TO 25001
      END
```

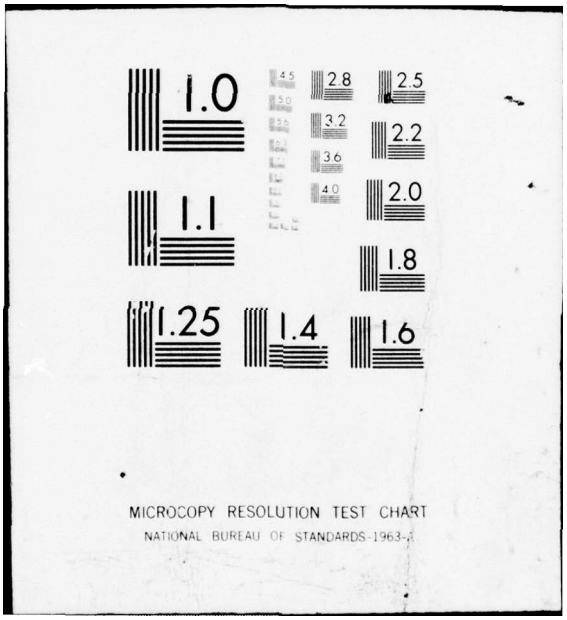
```
      SUBROUTINE PHONEY
      INTEGER FLGSPC,FLAGSP
      COMMON/LVVTR1/LVVSZE,LVVGSP,NODSPC(1000)/LVVTR2/LSTSPC(1000)
      COMMON/LVVTR3/LNKSPC(1000)/LVVTR9/FLGSPC(1000)
      COMMON/LVVTR5/LVFILE,LVCMPR,NODESP(     1)
     +/LVVTR6/LISTSP(     1)/LVVTR7/LINKSP(     1)
     +/LVVTR8/FLAGSP(     1)
      COMMON/LVRAND/LNKPRM,LVKS,LVKX,LVKDY,LVKDX,LVTEMP
      COMMON/LVARGS/LVFUNC,LVVARG,LVVAD,LVVPOS,LVVTYP,LVVAL,
     +LVHEAD,LVVNVL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP
      COMMON/LVTABL/LVTSIZ,LVMAP(     1)/LVVSEQ/LVSIZE,LVSQSP(     1)
C     EXECUTE
      LVVSZE=1000
      LVFILE=  0
      LVCMPR= 0
      LVSIZE=     1
      LVSKIP=1
      LNKPRM=17
      CALL LVSETP
      RETURN
      END
```

AD-A043 924 DAVID W TAYLOR NAVAL SHIP RESEARCH AND DEVELOPMENT CE--ETC F/G 9/2
MAINTENANCE MANUAL FOR AUDIT, A SYSTEM FOR ANALYZING SESCOMP SO--ETC(U)
AUG 77 R J WYBRANIEC, R REGEN

UNCLASSIFIED DTNSRDC-77-0075-VOL-3 NL

2 OF 2
AD
A043924

END
DATE FILMED
10-77
DDC


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
      SUBROUTINE PRNTS
      COMMON/LVARGS/LVFUNC,LVVARG,LVVAD,LVVPOS,LVVTYP,LVVAL,
     +LVHEAD,LVVNVL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP
      COMMON/LVTABL/LVTSIZ,LVMAP(     1)/LVVSEQ/LVSIZE,LVSQSP(      1)
      COMMON /HL/ HOL,ACTION,FUNC1,FUNC2,FUNC3,LEFT,RIGHT,STRING
      COMMON /VAR/ VFOR,NCHAR,NCHARP,CHAR,NDICT
      COMMON /TYP/ NARRAY,TYPE1,TYPE2,ERRFLG
      COMMON /STRING/ NTYPE,NSTR,STR
      COMMON /GIRL/ NNN(19),OPRAND
      LOGICAL ERRFLG
      INTEGER VFOR(15),CHAR,STRING,HOL,RIGHT,STR(1),OPRAND
C     EXECUTE
      GO TO 25000
25001 CONTINUE
      NCHAR=0
      NCHARP=0
      DO 5 I=1,15
    5 VFOR(I)=0
      NINT=1
      NINP=1
      DO 10 I=1,NSTR
C     STRING(+HOL,I @NODE,+STRING,I @NI=OPRAND/16)
      LVI     AAD =      STRING
C**** LVI     AAD              +          HOL
      LVVPOS =            I
      LVVTYP =    3
      LVFUNC=          HOL
      LVVARG= LVI     AAD
      CALL LVFIND(LV2       A,LV2       B,LV2       C,LV2       D)
      LVIAAI=LVIAAD
      IF(LVVAL .NE. -1) LVIAAI=LVVAL
C**** LVI     AAE              @          NODE
      NODE=LVIAAI
C**** LVI     AAD              +       STRING
      LVVPOS =            1
      LVVTYP =    3
      LVFUNC=      STRING
      LVVARG= LVI     AAD
      CALL LVFIND(LV2       E,LV2       F,LV2       G,LV2       H)
      LVIAAI=LVIAAD
      IF(LVVAL .NE. -1) LVIAAI=LVVAL
C**** LVI     AAE              @           NI
      NI=LVIAAI
C****           NI             =       OPRAND
      LVVAL = -100
      IF (         NI.NE.     OPRAND) LVVAL = -1
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          16
      NINT=NINT+1
   16 J=0
   20 J=J+1
C     NODE+LEFT.J/30 (+HOL.1 @CHAR,+HOL.2 @NDICT)
C**** NODE              +         LEFT
      LVVPOS =            J
      LVVTYP =    3
      LVFUNC=        LEFT
      LVVARG=        NODE
```

```
      CALL LVFIND(LV2       I,LV2       J,LV2       K,LV2       L)
      LVI    AAD =        NODE
      IF (LVVAL.NE.-1) LVI     AAD = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          30
      LVIAAI=LVIAAD
C     LVI    AAE              +          HOL
      LVVPOS =          1
      LVVTYP =   3
      LVFUNC=         HOL
      LVVARG=LVIAAI
      CALL LVFIND(LV2       M,LV2       N,LV2       O,LV2       P)
      LVIAAJ=LVIAAI
      IF(LVVAL .NE. -1) LVIAAJ=LVVAL
C     LVI    AAF              @          CHAR
      CHAR=LVIAAJ
      LVIAAJ=LVIAAD
C     LVI    AAF              +          HOL
      LVVPOS =          2
      LVVTYP =   3
      LVFUNC=         HOL
      LVVARG=LVIAAJ
      CALL LVFIND(LV2       Q,LV2       R,LV2       S,LV2       T)
      LVIAAI=LVIAAJ
      IF(LVVAL .NE. -1) LVIAAI=LVVAL
C     LVI    AAE              @          NDICT
      NDICT=LVIAAI
      CALL FORM
      GO TO 20
   30 CONTINUE
      IF(NINT .GT. NTMP) GO TO 35
C     NI(+HOL.1 @CHAR,+HOL.2 @NDICT)
      LVI    AAD =            NI
C     LVI    AAD              +          HOL
      LVVPOS =          1
      LVVTYP =   3
      LVFUNC=         HOL
      LVVARG= LVI    AAD
      CALL LVFIND(LV2       U,LV2       V,LV2       W,LV2       X)
      LVIAAI=LVIAAD
      IF(LVVAL .NE. -1) LVIAAI=LVVAL
C     LVI    AAE              @          CHAR
      CHAR=LVIAAI
C     LVI    AAD              +          HOL
      LVVPOS =          2
      LVVTYP =   3
      LVFUNC=         HOL
      LVVARG= LVI    AAD
      CALL LVFIND(LV2       Y,LV2       Z,LV2       0,LV2       1)
      LVIAAI=LVIAAD
      IF(LVVAL .NE. -1) LVIAAI=LVVAL
C     LVI    AAE              @          NDICT
      NDICT=LVIAAI
      CALL FORM
      GO TO 37
   35 NTMP=NINT
C     NODE+HOL @NDICT
```

```
C****       NODE          +          HOL
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=        HOL
      LVVARG=       NODE
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1    AAD =          NODE
      IF (LVVAL.NE.-1) LV1    AAD = LVVAL
C**** LV1    AAD           @         NDICT
           NDICT = LV1    AAD
C     OPRAND+HOL @CHAR
C****      OPRAND           +          HOL
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=         HOL
      LVVARG=     OPRAND
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1    AAD =      OPRAND
      IF (LVVAL.NE.-1) LV1    AAD = LVVAL
C**** LV1    AAD           @         CHAR
           CHAR = LV1    AAD
      CALL FORM
   37 J=0
   40 J=J+1
C     NODE+RIGHT.J/10 (+HOL.1 @CHAR,+HOL.2 @NDICT)
C****      NODE           +         RIGHT
      LVVPOS =           J
      LVVTYP =   3
      LVFUNC=       RIGHT
      LVVARG=        NODE
      CALL LVFIND(LV2       2,LV2       3,LV2       4,LV2       5)
      LV1    AAD =         NODE
      IF (LVVAL.NE.-1) LV1    AAD = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          10
      LV1AAI=LV1AAD
C**** LV1    AAE           +          HOL
      LVVPOS =           1
      LVVTYP =   3
      LVFUNC=         HOL
      LVVARG=LV1AAI
      CALL LVFIND(LV2       6,LV2       7,LV2       8,LV2       9)
      LV1AAJ=LV1AAI
      IF(LVVAL .NE. -1) LV1AAJ=LVVAL
C**** LV1    AAF           @         CHAR
      CHAR=LV1AAJ
      LV1AAJ=LV1AAD
C**** LV1    AAF           +          HOL
      LVVPOS =          2
      LVVTYP =   3
      LVFUNC=         HOL
      LVVARG=LV1AAJ
      CALL LVFIND(LV2      AA,LV2      AB,LV2      AC,LV2      AD)
      LV1AAI=LV1AAJ
      IF(LVVAL .NE. -1) LV1AAI=LVVAL
```

```
C**** LVI    AAE            @       NDICT
      NDICT=LVIAAI
      CALL FORM
      GO TO 40
   10 CONTINUE
      NC=1+(NCHARP-1)/8
  100 FORMAT(1X,15A8)
      IF(ERRFLG) PRINT 100,(VFOR(I),I=1,NC)
C     COMPLETE
      RETURN
25000 CONTINUE
      LV2A=0
      LV2B=0
      LV2C=0
      LV2D=0
      LV2E=0
      LV2F=0
      LV2G=0
      LV2H=0
      LV2I=0
      LV2J=0
      LV2K=0
      LV2L=0
      LV2M=0
      LV2N=0
      LV2O=0
      LV2P=0
      LV2Q=0
      LV2R=0
      LV2S=0
      LV2T=0
      LV2U=0
      LV2V=0
      LV2W=0
      LV2X=0
      LV2Y=0
      LV2Z=0
      LV20=0
      LV21=0
      LV22=0
      LV23=0
      LV24=0
      LV25=0
      LV26=0
      LV27=0
      LV28=0
      LV29=0
      LV2AA=0
      LV2AB=0
      LV2AC=0
      LV2AD=0
      GO TO 25001
      END
```

```
      SUBROUTINE PROG                                                    PROG   2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),   RICH   2
     9 JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                   RICH   4
      IDTYP=2
      CALL STORE                                                         PROG  14
      RETURN                                                             PROG  21
      END                                                                PROG  22
```

35 Bits

```
REAL FUNCTION Q1REAL(X)
DATA R/O7777777777776/
Q1REAL=AND(X,R)
RETURN
END
DOUBLE PRECISION FUNCTION Q1DPRE(X)
DOUBLE PRECISION X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/O7777777777776/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1DPRE=T
RETURN
END
COMPLEX FUNCTION Q1COMP(X)
COMPLEX X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/O7777777777776/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1COMP=T
RETURN
END
```

34 Bits

```
REAL FUNCTION Q1REAL(X)
DATA R/O7777777777774/
Q1REAL=AND(X,R)
RETURN
END
DOUBLE PRECISION FUNCTION Q1DPRE(X
DOUBLE PRECISION X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/O7777777777774/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1DPRE=T
RETURN
END
COMPLEX FUNCTION Q1COMP(X)
COMPLEX X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/O7777777777774/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1COMP=T
RETURN
END
```

33 Bits

```
REAL FUNCTION Q1REAL(X)
DATA R/07777777777770/
Q1REAL=AND(X,R)
RETURN
END
DOUBLE PRECISION FUNCTION Q1DPRE(X)
DOUBLE PRECISION X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/07777777777770/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1DPRE=T
RETURN
END
COMPLEX FUNCTION Q1COMP(X)
COMPLEX X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/07777777777770/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1COMP=T
RETURN
END
```

32 Bits

```
REAL FUNCTION Q1REAL(X)
DATA R/0777777777760/
Q1REAL=AND(X,R)
RETURN
END
DOUBLE PRECISION FUNCTION Q1DPRE(X)
DOUBLE PRECISION X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/0777777777760/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1DPRE=T
RETURN
END
COMPLEX FUNCTION Q1COMP(X)
COMPLEX X,T
DIMENSION W(2)
EQUIVALENCE(T,W(1))
DATA R/0777777777760/
T=X
W(1)=AND(W(1),R)
W(2)=AND(W(2),R)
Q1COMP=T
RETURN
END
```

31 Bits

```
      REAL FUNCTION Q1REAL(X)
      DATA R/O777777777740/
      Q1REAL=AND(X,R)
      RETURN
      END
      DOUBLE PRECISION FUNCTION Q1DPRE(X)
      DOUBLE PRECISION X,T
      DIMENSION W(2)
      EQUIVALENCE(T,W(1))
      DATA R/O777777777740/
      T=X
      W(1)=AND(W(1),R)
      W(2)=AND(W(2),R)
      Q1DPRE=T
      RETURN
      END
      COMPLEX FUNCTION Q1COMP(X)
      COMPLEX X,T
      DIMENSION W(2)
      EQUIVALENCE(T,W(1))
      DATA R/O777777777740/
      T=X
      W(1)=AND(W(1),R)
      W(2)=AND(W(2),R)
      Q1COMP=T
      RETURN
      END
```

30 Bits

```
      REAL FUNCTION Q1REAL(X)
      DATA R/O777777777700/
      Q1REAL=AND(X,R)
      RETURN
      END
      DOUBLE PRECISION FUNCTION Q1DPRE(X)
      DOUBLE PRECISION X,T
      DIMENSION W(2)
      EQUIVALENCE(T,W(1))
      DATA R/O777777777700/
      T=X
      W(1)=AND(W(1),R)
      W(2)=AND(W(2),R)
      Q1DPRE=T
      RETURN
      END
      COMPLEX FUNCTION Q1COMP(X)
      COMPLEX X,T
      DIMENSION W(2)
      EQUIVALENCE(T,W(1))
      DATA R/O777777777700/
      T=X
      W(1)=AND(W(1),R)
      W(2)=AND(W(2),R)
      Q1COMP=T
      RETURN
      END
```

```
      SUBROUTINE REALCK                                                 REAL   2
      COMMON A(1324),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      COMMON/LOGIC/LOG,LOGST                                            REAL   4
      COMMON/REALND/IREAL,IRELND,IP                                     REAL   5
      INTEGER A,DECPT,EEE,PLUS,MINUS,DEE,ELOC                           REAL   6
      DATA DECPT/1H./,EEE/1HE/, PLUS/1H+/, MINUS/1H-/,DEE/1HD/          REAL   7
      IDES=0                                                            REAL   8
      IRELND=0                                                          REAL   9
      IF(IP .GE. N) GO TO 90                                            REAL  10
      IF(NEXT(IP) .EQ. DECPT) GO TO 5                                   REAL  11
      IF(ITYPE(IP) .EQ. 2) GO TO 10                                     REAL  12
      GO TO 90                                                          REAL  13
    5 IF(JPTR .GT. N) GO TO 90                                          REAL  14
      IF(ITYPE(JPTR) .NE. 2) GO TO 90                                   REAL  15
      GO TO 20                                                          REAL  16
   10 IF(JPTR .GT. N) GO TO 90                                          REAL  17
   12 IF(ITYPE(JPTR) .EQ. 2) GO TO 15                                   REAL  18
      IF(A(JPTR-1) .NE. DECPT) GO TO 90                                 REAL  19
      LOGST=JPTR                                                        REAL  20
      CALL LOGCHK                                                       REAL  21
      IF(LOG .EQ. 1) GO TO 90                                           REAL  22
      JPTR=LOGST                                                        REAL  23
      GO TO 20                                                          REAL  24
   15 IF(JPTR .GT. N) GO TO 90                                          REAL  25
      GO TO 12                                                          REAL  26
   20 IREAL=1                                                           REAL  27
      IF(JPTR .GT. N) GO TO 35                                          REAL  28
   22 IF(ITYPE(JPTR) .EQ. 2) GO TO 25                                   REAL  29
      IF(A(JPTR-1) .EQ. EEE) GO TO 24                                   REAL  30
      IF(A(JPTR-1) .NE. DEE) GO TO 30                                   REAL  31
      IDES=1                                                            REAL  32
   24 ELOC=JPTR-2                                                       REAL  33
      GO TO 90                                                          REAL  34
   25 IF(JPTR .GT. N) GO TO 35                                          REAL  35
      GO TO 22                                                          REAL  36
   30 ELOC=JPTR-2                                                       REAL  37
   32 IRELND=ELOC                                                       REAL  38
      RETURN                                                            REAL  39
   35 IRELND=N                                                          REAL  40
      RETURN                                                            REAL  41
   90 IF(JPTR .GT. N) GO TO 32                                          REAL  42
      NXT=NEXT(JPTR)                                                    REAL  43
      IF(NXT .EQ. PLUS .OR. NXT .EQ. MINUS) GO TO 95                    REAL  44
      IF(ITYPE(JPTR-1) .NE. 2) GO TO 32                                 REAL  45
      IF(JPTR .GT. N) GO TO 35                                          REAL  46
      GO TO 97                                                          REAL  47
   95 IF(JPTR .GT. N) GO TO 32                                          REAL  48
      IF(ITYPE(JPTR) .NE. 2) GO TO 32                                   REAL  49
   97 IF(ITYPE(JPTR) .EQ. 2) GO TO 50                                   REAL  50
      IRELND=JPTR-2                                                     REAL  51
      RETURN                                                            REAL  52
   50 IF(JPTR .GT. N) GO TO 35                                          REAL  53
      GO TO 97                                                          REAL  54
   90 IREAL=0                                                           REAL  55
      RETURN                                                            REAL  56
      END                                                               REAL  57
```

```
      SUBROUTINE RECOG(FIN)
      COMMON/LVARGS/LVFUNC,LVVARG,LVVAD,LVVPOS,LVVTYP,LVVAL,
     +LVHEAD,LVVNVL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP
      COMMON/LVTABL/LVTSIZ,LVMAP(    1)/LVVSEG/LVSIZE,LVSGSP(    1)
      COMMON /HL/ HOL,ACTION,FUNC1,FUNC2,FUNC3,LEFT,RIGHT,STRING,MAXJ
      COMMON /NEED/ START,ASSOC,LEVEL,STOP
      COMMON /STRING/ NNN(2),STR
      COMMON/NEEDS/STJ,JSTACK,R,JAS,J,JLAST,RTEMP,STACK(900,4)
      INTEGER HOL
      INTEGER START,ASSOC,STOP,RETRN,R,STJ,STACK,STR(1),ACTION,STRING,
     , RTEMP
      LOGICAL FAIL,FIN
      FIN=.FALSE.
C     EXECUTE
      GO TO 25000
25001 CONTINUE
      JSTACK=0
      J=1
C     START @R
C****       START          @          R
                R =      START
C     STRING+STRING,J/70 @STJ
C****       STRING           +      STRING
      LVVPOS =          J
      LVVTYP =   3
      LVFUNC=     STRING
      LVVARG=     STRING
      CALL LVFIND(LV2      A,LV2      B,LV2      C,LV2      D)
      LVI    AAD =     STRING
      IF (LVVAL.NE.-1) LVI    AAD = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          70
C**** LVI    AAD           @          STJ
             STJ = LVI    AAD
      M=-1
C     STRING+HOL,1 STRING @@M@@
C****       STRING           +          HOL
      LVVPOS =          1
      LVVTYP =   3
      LVFUNC=         HOL
      LVVARG=     STRING
      CALL LVFIND(LV2      E,LV2      F,LV2      G,LV2      H)
      LVI    AAD =     STRING
      IF (LVVAL.NE.-1) LVI    AAD = LVVAL
C**** LVI    AAD      STRING           @@
      LVDEST=  0
      LVIAAH=M
      LVTYPE(1) =  1
      LVVALS(1)=LVIAAH
      LVDEST=  0
      LVVNVL =  1
      LVFUNC =     STRING
      LVVARG=LVI    AAD
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
    6 CONTINUE
C   10 R(+ASSOC//15,+STOP//15,=STOP/20)
   10 CONTINUE
      LVI    AAD =          R
C**** LVI    AAD           +       ASSOC
```

```
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=       ASSOC
      LVVARG= LVI     AAD
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LVIAAI=LVIAAD
      IF(LVVAL .NE. -1) LVIAAI=LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO          15
C     LVI     AAD             .         STOP
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=       STOP
      LVVARG= LVI     AAD
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LVIAAI=LVIAAD
      IF(LVVAL .NE. -1) LVIAAI=LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO          15
C     LVI     AAD             =         STOP
      LVVAL = -100
      IF (LVI    AAD.NE.      STOP) LVVAL = -1
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          20
   15 JSTACK=JSTACK+1
      STACK(JSTACK,1)=R
      STACK(JSTACK,2)=0
      STACK(JSTACK,3)=J
      STACK(JSTACK,4)=0
C  20 R+ACTION/22 @N
   20 CONTINUE
C     R             +       ACTION
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=     ACTION
      LVVARG=           R
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LVI     AAD =           R
      IF (LVVAL.NE.-1) LVI     AAD = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          22
C     LVI     AAD             @           N
                N = (VI     AAD
      CALL SEMANT(N,FAIL)
      IF(FAIL) GO TO 99
      GO TO 25
C  22 R+STJ/99 @R
   22 CONTINUE
C     R             +         STJ
      LVVTYP = 3
      LVVPOS = 1
```

```
      LVINDX = 0
      LVFUNC=         STJ
      LVVARG=           R
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LVI     AAD =             R
      IF (LVVAL.NE.-1) LVI     AAD = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          99
C**** LVI     AAD             @           R
                R = LVI     AAD
   25 J=J+1
      IF(J .GT. MAXJ) MAXJ=J
C  30 STRING+STRING.J @STJ//A
   30 CONTINUE
C****      STRING            +       STRING
      LVVPOS =            J
      LVVTYP =  3
      LVFUNC=      STRING
      LVVARG=      STRING
      CALL LVFIND(LV2       I,LV2       J,LV2       K,LV2       L)
      LVI     AAD =       STRING
      IF (LVVAL.NE.-1) LVI     AAD = LVVAL
C**** LVI     AAD             @           STJ
                STJ = LVI     AAD
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO          A
   40 STJ=-1
C     R+ACTION/42 @N
C****           R                   ACTION
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=      ACTION
      LVVARG=           R
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LVI     AAD =             R
      IF (LVVAL.NE.-1) LVI     AAD = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          42
C**** LVI     AAD             @           N
                N = LVI     AAD
      CALL SEMANT(N,FAIL)
      IF(FAIL) GO TO 99
   42 CALL SSTOP(FAIL)
      IF(FAIL) GO TO 99
      JSTACK=JSTACK+1
      STACK(JSTACK,1)=R
      STACK(JSTACK,2)=0
      STACK(JSTACK,3)=J
      STACK(JSTACK,4)=0
C     R+ACTION/99 @N
C****           R           +       ACTION
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
```

```
179*            LVFUNC=      ACTION
180*            LVVARG=            R
181*            CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
182*            LVI     AAD =           R
183*            IF (LVVAL.NE.-1) LVI     AAD = LVVAL
184*            LVVTR = LVVAL
185*            LVVAL = -100
186*            IF (LVVTR.EQ.-1) GO TO         44
187*      C**** LVI     AAD            @            N
188*                    N = LVI     AAD
189*            CALL SEMANT(N,FAIL)
190*            IF(FAIL) GO TO 99
191*         44 CALL SLEVEL(FAIL)
192*            IF(FAIL) RETURN
193*            GO TO 40
194*         99 CONTINUE
195*            CALL RECOV(RETRN)
196*            IF(RETRN .LT. 0) GO TO 70
197*      C     RETRN=ASSOC//30
198*      C**** RETRN            =        ASSOC
199*            LVVAL = -100
200*            IF (     RETRN.NE.     ASSOC) LVVAL = -1
201*            LVVTR = LVVAL
202*            LVVAL = -100
203*            IF (LVVTR.NE.-1) GO TO          30
204*            IF(RETRN .EQ. 0) GO TO 10
205*            CALL SLEVEL(FAIL)
206*            IF(FAIL) GO TO 65
207*            GO TO 30
208*         65 IF(JSTACK .LE. 1) GO TO 70
209*            JSTACK=JSTACK-1
210*            GO TO 99
211*         70 FIN=.TRUE.
212*            RETURN
213*      C     COMPLETE
214*      25000 CONTINUE
215*            LV2A=0
216*            LV2B=0
217*            LV2C=0
218*            LV2D=0
219*            LV2E=0
220*            LV2F=0
221*            LV2G=0
222*            LV2H=0
223*            LV2I=0
224*            LV2J=0
225*            LV2K=0
226*            LV2L=0
227*            GO TO 25001
228*            END
```

```
      SUBROUTINE RECOV(RETRN)
      COMMON/LVARGS/LVFUNC,LVVARG,LVVAD,LVVPOS,LVVTYP,LVVAL,
     +LVHEAD,LVVNVL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP
      COMMON/LVTABL/LVTSIZ,LVMAP(     1)/LVVSEQ/LVSIZE,LVSQSP(     1)
      COMMON /NEED/ START,ASSOC,LEVEL,STOP
      COMMON/NEEDS/STJ,JSTACK,R,JAS,J,JLAST,RTEMP,STACK(400,4)
      COMMON /STRING/ NNN(2),STR
      COMMON /HL/ HOL,ACTION,FUNC1,FUNC2,FUNC3,LEFT,RIGHT,STRING
      INTEGER START,ASSOC,STOP,STACK,STR(1),STJ,R,STRING,RETRN,TEMP
     +,RIGHT,HOL
      LOGICAL FAIL
C     EXECUTE
      GO TO 25000
25001 CONTINUE
   10 R=STACK(JSTACK,1)
      JAS=STACK(JSTACK,2)+1
C     R+ASSOC.JAS @TEMP//30
C**** R + ASSOC
      LVVPOS = JAS
      LVVTYP = 3
      LVFUNC= ASSOC
      LVVARG= R
      CALL LVFIND(LV2 A,LV2 B,LV2 C,LV2 D)
      LV1 AAD = R
      IF (LVVAL.NE.-1) LV1 AAD = LVVAL
C**** LV1 AAD @ TEMP
      TEMP = LV1 AAD
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO 30
C  15 R(=STOP//40,+STOP/16=STOP//40 @R)
   15 CONTINUE
      LV1 AAD = R
C**** LV1 AAD = STOP
      LVVAL = -100
      IF (LV1 AAD.NE. STOP) LVVAL = -1
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO 40
C**** LV1 AAD + STOP
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC= STOP
      LVVARG= LV1 AAD
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LV1AAH=LV1AAD
      IF(LVVAL .NE. -1) LV1AAH=LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO 16
C**** LV1 AAF = STOP
      LVVAL = -100
      IF(LV1AAH .NE. STOP) LVVAL=-1
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO 40
C**** LV1 AAF @ R
      R=LV1AAH
      J=STACK(JSTACK,3)
      JSTACK=JSTACK-1
      RETRN=0
      RETURN
   16 JSTACK=JSTACK-1
      IF(JSTACK .LE. 0) GO TO 20
      IF(STACK(JSTACK,3) .LT. J) CALL SEMANT(0,FAIL)
      J=STACK(JSTACK,3)
      GO TO 10
   20 RETRN=-1
      RETURN
```

```
      40 CONTINUE
         J=STACK(JSTACK,3)
         IF(STACK(JSTACK,4) .GT. 0) GO TO 16
         RETRN=STOP
         RETURN
C     30 TEMP @R
      30 CONTINUE
C****       TEMP          @          R
                R =       TEMP
         IF(JSTACK .EQ. 1) GO TO 35
         IF( R .NE. STACK(JSTACK,1)) GO TO 35
         NTEMP=STACK(JSTACK-1,1)
         JMARK=JSTACK
      31 STACK(JMARK,4)=-1
         JMARK=JMARK-1
         IF(R .EQ. STACK(JMARK,1) .AND. STACK(JMARK,4) .LT. 0) GO TO 15
         IF(R .EQ. STACK(JMARK,1) .AND. JMARK .NE. 0) GO TO 31
         IF(R .NE. NTEMP .OR. JAS .NE. STACK(JSTACK-1,2) .OR.
        + STACK(JSTACK-1,4) .GE. 0) GO TO 35
         GO TO 15
      35 CONTINUE
         IF(STACK(JSTACK,3) .LT. J) CALL SEMANT(0,FAIL)
         STACK(JSTACK,2)=JAS
         J=STACK(JSTACK,3)
         IF(STACK(JSTACK,4) .GT. 0) STACK(JSTACK,4)=0
         RETRN=ASSOC
         RETURN
C     COMPLETE
25000 CONTINUE
         LV2A=0
         LV2B=0
         LV2C=0
         LV2D=0
         GO TO 25001
         END
```

```
      SUBROUTINE ROLCHK(I1,I2,I3,I4,I5,I6)
      DIMENSION IARG(6)
      IARG(1)=FLD(0,6,I1)
      IARG(2)=FLD(0,6,I2)
      IARG(3)=FLD(0,6,I3)
      IARG(4)=FLD(0,6,I4)
      IARG(5)=FLD(0,6,I5)
      IARG(6)=FLD(0,6,I6)
      ISUBNM=0
      DO 10 I=1,6
      IBIT=6*(I-1)
   10 FLD(IBIT,6,ISUBNM)=IARG(I)
      WRITE(3) ISUBNM
      RETURN
      END
```

```
      SUBROUTINE SEARCH                                                 SEARCH 2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      DO 20 IDTYP=1,2                                                   SEARCH 4
      J=INITID(IDTYP)                                                   SEARCH 5
      IF(J .EQ. 0) GO TO 15                                             SEARCH 6
      DO 10 I=1,NID                                                     SEARCH 7
      IF(IDTBL(1,J) .NE. NXTID) GO TO 5                                 SEARCH 8
      ISRCH(IDTYP)=1                                                    SEARCH 9
      IDES=LOC
      LOC=J                                                             SEARCH10
      GO TO 20                                                          SEARCH11
    5 J=IDTBL(2,J)                                                      SEARCH12
      IF(J .EQ. 0) GO TO 15                                             SEARCH13
   10 CONTINUE                                                          SEARCH14
   15 ISRCH(IDTYP)=0                                                    SEARCH15
   20 CONTINUE                                                          SEARCH16
      RETURN                                                            SEARCH17
      END                                                               SEARCH18
```

```
      SUBROUTINE SEMANT(N,FAIL)
      COMMON/LVARGS/LVFUNC,LVVARG,LVVAD,LVVPOS,LVVTYP,        LVVAL,
     +LVHEAD,LVVNVL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP
      COMMON/LVTABL/LVTSIZ,LVMAP(     1)/LVVSEQ/LVSIZE,LVSQSP(      1)
      COMMON/FUNC/  NARY(5,17),MARGS,IARGS(50),FNCLOC(5),NFUNC
      COMMON/HL/HOL,ACTION,FUNC1,FUNC2,FUNC3,LEFT,RIGHT,STRING,MAXJ
      COMMON /TYP/ NARRAY,TYPE1,TYPE2,ERRFLG
      COMMON /STRING/ NTYPE,NSTR,STR
      COMMON /JL/ JSTOP
      COMMON /GIRL/NTERMS,PLUS,MINUS,SLASH,LPAR,RPAR,COMMA,STAR,EXP,LT,
     +LE,GT,GE,EQ,NE,OR,AND,NOT,EQUALS,OPRAND
      COMMON/NEEDS/STJ,JSTACK,R,JAS,J,JLAST,RTEMP,STACK(400,4)
      COMMON /NEED/ START,ASSOC,LEVEL,STOP
      COMMON/NOPAR/NOPAR,NDEP,NDEPTH,NFLAG
      INTEGER HOL,ACTION,FUNC,LEFT,RIGHT,STRING,RPAR,STJ,R,STACK
     +,EXP,FUNC1,FUNC2,FUNC3,TYPE1,TYPE2,TYPE(5),STR(1),STOP
     $ ,ALPHA,BETA,GAMMA,OPRAND,EQUALS,AND,OR,COMMA
      LOGICAL SKIP,FLAG,ERRFLG,FAIL,NOTFLG
      INTEGER FUNCRF,ZERO,BITPUT,PLUS,FL(3)
      INTEGER GETTYP,GETDIM
      DATA FLAG/.FALSE./,FUNCRF/86/,ZERO/0/
      DATA (TYPE(I),I=1,5)/4HREAL,6HCOMPLX,6HDOUBLE,6HINTEGR,6HLOGICL/
      GETTYP(II)=MOD(II,100000)/10000
      GETDIM(II)=MOD(II,1000000)/100000
C     EXECUTE
      GO TO 25000
25001 CONTINUE
      FAIL=.FALSE.
      IF(N .EQ. 0) GO TO 999
      GO TO(10,20,30,40,50,60,70,80,90,1000,1100,1200,1300,1400,1500,
     $ 1600),N
C  10 R=STJ/11 @R//12
   10 CONTINUE
C****           R             +          STJ
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC=          STJ
      LVVARG=            R
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LVI    AAD =           R
      IF (LVVAL.NE.-1) LVI    AAD = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          11
C**** LVI    AAD             @            R
                R = LVI    AAD
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO          12
   11 FAIL=.TRUE.
      RETURN
C PRIMARY RECOGNIZED
   12 IF(STJ .EQ. PLUS .OR. STJ .EQ. MINUS) GO TO 126
      IF(STJ .NE. RPAR) GO TO 121
      JSTACK=JSTACK+1
      STACK(JSTACK,1)=STOP
      STACK(JSTACK,2)=0
      STACK(JSTACK,3)=J
      STACK(JSTACK,4)=0
      NTMP=R
      CALL SLEVEL(SKIP)
      JSTACK=JSTACK-1
      R=NTMP
      JLAST=1
      IF(JSTOP .GT. 0) JLAST=STACK(JSTOP,3)
C     STRING=HOL,JLAST(-STRING,STRING @@TYPE(@@)
```

```
C****      STRING              +          HOL
          LVVPOS =       JLAST
          LVVTYP =    3
          LVFUNC=          HOL
          LVVARG=      STRING
          CALL LVFIND(LV2       A,LV2       B,LV2       C,LV2       D)
          LVI     AAD =       STRING
          IF (LVVAL.NE.-1) LVI     AAD = LVVAL
          LVI     AAH = LVI     AAD
C****     LVI     AAF             =       STRING
          LVVAD=-1
          LVVTYP=-1
          LVVPOS=1
          LVFUNC=    STRING
          LVVARG=LVI     AAH
          CALL LVDLET
          LVI     AAH = LVI     AAD
C****     LVI     AAF       STRING          BB
          LVDEST=  0
          LVI     AAI = TYPEI
          LVTYPE(1) =  1
          LVVALS(1) = LVI     AAI
          LVDEST=  0
          LVVNVL =  1
          LVFUNC =      STRING
          LVVARG=LVI     AAH
          CALL LVNSRT
          IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
          IF(LVVAL.LT.0) RETURN
          RETURN
      121 CONTINUE
C  GET TYPE
          BETA=GETDIM(STR(J))
          IF(BETA .NE. 5) GO TO 125
C  OPERAND IS A FUNCTION REFERENCE
          IF(NDEP .EQ. 0) GO TO 18
          LVVPOS=-LVVPOS
          LVVTYP=  3
          LVVPOS=          1
          LVDEST=  2
          LVI     AAD = 1
          LVTYPE(1) =  1
          LVVALS(1) = LVI     AAD
          LVDEST=  2
          LVVNVL =  1
          LVFUNC =       FUNC1
          LVVARG=     OPRAND
          CALL LVNSRT
          IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
          IF(LVVAL.LT.0) RETURN
       18 R=FUNCRF
          JSTACK=JSTACK+1
          STACK(JSTACK,1)=R
          STACK(JSTACK,2)=0
          STACK(JSTACK,3)=J+1
          STACK(JSTACK,4)=0
      125 ALPHA=GETTYP(STR(J))
          IF(TYPE1 .GE. 0) GO TO 13
```

```
126•      C  SET TYPE OF STATEMENT
127•             TYPE1=ALPHA
128•             IF(NTYPE .EQ. 3) TYPE1=-1
129•      C 126 STRING+HOL.J(-STRING,STRING @@TYPE1@@)
130•        126 CONTINUE
131•      C••••      STRING              +          HOL
132•             LVVPOS =            J
133•             LVVTYP =   3
134•             LVFUNC=          HOL
135•             LVVARG=      STRING
136•             CALL LVFIND(LV2       E,LV2       F,LV2       G,LV2       H)
137•             LV1     AAH =      STRING
138•             IF (LVVAL.NE.-1) LV1     AAH = LVVAL
139•             LV1     AAJ = LV1     AAH
140•      C•••• LV1     AAF                    STRING
141•             LVVAD=-1
142•             LVVTYP=-1
143•             LVVPOS=1
144•             LVFUNC=     STRING
145•             LVVARG=LV1     AAJ
146•             CALL LVDLET
147•             LV1     AAJ = LV1     AAH
148•      C•••• LV1     AAF       STRING            @@
149•             LVDEST=  0
150•             LV1     AAK = TYPE1
151•             LVTYPE(1) =  1
152•             LVVALS(1) = LV1     AAK
153•             LVDEST=  0
154•             LVVNVL =  1
155•             LVFUNC =      STRING
156•             LVVARG=LV1     AAJ
157•             CALL LVNSRT
158•             IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
159•             IF(LVVAL.LT.0) RETURN
160•             RETURN
161•          13 IF(FLAG) GO TO 15
162•      C   CHECK FOR MIXED MODE EXPRESSION
163•             IF(TYPE1 .EQ. ALPHA .OR. ALPHA .EQ. 5) GO TO 16
164•          14 N1=TYPE1+1
165•             N2=ALPHA+1
166•             ERRFLG=.TRUE.
167•             CALL ERROR(77,TYPE(N1),TYPE(N2))
168•      C      STRING+HOL.J(-STRING,STRING @@TYPE1@@)
169•      C••••      STRING              +          HOL
170•             LVVPOS =            J
171•             LVVTYP =   3
172•             LVFUNC=          HOL
173•             LVVARG=      STRING
174•             CALL LVFIND(LV2       I,LV2       J,LV2       K,LV2       L)
175•             LV1     AAH =      STRING
176•             IF (LVVAL.NE.-1) LV1     AAH = LVVAL
177•             LV1     AAJ = LV1     AAH
178•      C•••• LV1     AAF              -       STRING
179•             LVVAD=-1
180•             LVVTYP=-1
181•             LVVPOS=1
182•             LVFUNC=     STRING
183•             LVVARG=LV1     AAJ
```

```
184•            CALL LVDLET
185•            LV1    AAJ = LV1    AAH
186•      C**** LV1    AAF       STRING           @@
187•            LVDEST=  0
188•            LV1    AAL = TYPE1
189•            LVTYPE(1) =  1
190•            LVVALS(1) = LV1    AAL
191•            LVDEST=  0
192•            LVVNVL =  1
193•            LVFUNC =     STRING
194•            LVVARG=LV1    AAJ
195•            CALL LVNSRT
196•            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
197•            IF(LVVAL.LT.0) RETURN
198•            RETURN
199•      C  PARSING  AN EXPONENT
200•         15 IF((ALPHA .EQ. 3 .AND. TYPE1 .EQ. 3).OR.((TYPE1 .EQ. 0 .OR. TYPE1
201•           + .EQ. 2) .AND.
202•           +(ALPHA .EQ. 0 .OR. ALPHA .EQ. 2))) GO TO 16
203•            CALL ERROR(78,J,KDM2)
204•            ERRFLG=.TRUE.
205•      C     STRING+HOL.J(-STRING,STRING @@TYPE1@@)
206•      C****     STRING                        HOL
207•            LVVPOS =          J
208•            LVVTYP =   3
209•            LVFUNC=        HOL
210•            LVVARG=     STRING
211•            CALL LVFIND(LV2      M,LV2      N,LV2      O,LV2      P)
212•            LV1    AAH =     STRING
213•            IF (LVVAL.NE.-1) LV1    AAH = LVVAL
214•            LV1    AAJ = LV1    AAH
215•      C**** LV1    AAF            =      STRING
216•            LVVAD=-1
217•            LVVTYP=-1
218•            LVVPOS=1
219•            LVFUNC=    STRING
220•            LVVARG=LV1    AAJ
221•            CALL LVDLET
222•            LV1    AAJ = LV1    AAH
223•      C**** LV1    AAF       STRING           @@
224•            LVDEST=  0
225•            LV1    AAM = TYPE1
226•            LVTYPE(1) =  1
227•            LVVALS(1) = LV1    AAM
228•            LVDEST=  0
229•            LVVNVL =  1
230•            LVFUNC =     STRING
231•            LVVARG=LV1    AAJ
232•            CALL LVNSRT
233•            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
234•            IF(LVVAL.LT.0) RETURN
235•            RETURN
236•         16 IF((.NOT. FLAG .AND. TYPE1 .LT. ALPHA).OR.(FLAG .AND. ALPHA .NE. 3
237•           + .AND. TYPE1 .LT. ALPHA)) TYPE1=ALPHA
238•      C     STRING+HOL.J(-STRING,STRING @@TYPE1@@)
239•      C****     STRING            +          HOL
240•            LVVPOS =          J
241•            LVVTYP =   3
```

```
            LVFUNC=          HOL
            LVVARG=      STRING
            CALL LVFIND(LV2       Q,LV2       R,LV2       S,LV2       T)
            LVI     AAH =      STRING
            IF (LVVAL.NE.-1) LVI     AAH = LVVAL
            LVI     AAJ = LVI     AAH
      C**** LVI     AAF             -       STRING
            LVVAD=-1
            LVVTYP=-1
            LVVPOS=1
            LVFUNC=     STRING
            LVVARG=LVI     AAJ
            CALL LVDLET
            LVI     AAJ = LVI     AAH
      C**** LVI     AAF        STRING            @@
            LVDEST=  0
            LVI     AAN = TYPEI
            LVITYPE(1) =  1
            LVVALS(1) = LVI     AAN
            LVDEST=  0
            LVVNVL =  1
            LVFUNC =      STRING
            LVVARG=LVI     AAJ
            CALL LVNSRT
            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
            IF(LVVAL.LT.0) RETURN
            RETURN
      C  WILL SCAN AN EXPONENT
         20 IF(STJ .LT. 0) RETURN
      C     R=STJ/11 @R
      C****           R             +          STJ
            LVVTYP = 3
            LVVPOS = 1
            LVINDX = 0
            LVFUNC=          STJ
            LVVARG=            R
            CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
            LVI     AAH =           R
            IF (LVVAL.NE.-1) LVI     AAH = LVVAL
            LVVTR = LVVAL
            LVVAL = -100
            IF (LVVTR.EQ.-1) GO TO          11
      C**** LVI     AAD             @            R
                  R = LVI     AAH
            FLAG=.TRUE.
            RETURN
      C  RECOGNIZED A TERM,PRODUCT OR PRIMARY PERHAPS NEEDING PARENTHESIZATION
         30 CONTINUE
            KTMP=R
            IF(NDEP .EQ. 0) GO TO 39
            LVVPOS=-LVVPOS
            LVVTYP=  3
            LVVPOS=          1
            LVDEST=  2
            LVI     AAH = 1
            LVITYPE(1) =  1
            LVVALS(1) = LVI     AAH
            LVDEST=  2
```

```
300*            LVVNVL =  1
301*            LVFUNC =      FUNC1
302*            LVVARG=    OPRAND
303*            CALL LVNSRT
304*            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
305*            IF(LVVAL.LT.0) RETURN
306*         39 CONTINUE
307*            ITEST=0
308*            IF(STJ .LT. 0) GO TO 31
309*      C     R*STJ @R//32
310*      C****           R            *          STJ
311*            LVVTYP = 3
312*            LVVPOS = 1
313*            LVINDX = 0
314*            LVFUNC=         STJ
315*            LVVARG=           R
316*            CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
317*            LVI     AAJ =           R
318*            IF (LVVAL.NE.-1) LVI     AAJ = LVVAL
319*      C**** LVI     AAF             @            R
320*                    R = LVI     AAJ
321*            LVVTR = LVVAL
322*            LVVAL = -100
323*            IF (LVVTR.NE.-1) GO TO          32
324*      C   31 R*STOP/39 @R
325*          31 CONTINUE
326*      C****           R            *          STOP
327*            LVVTYP = 3
328*            LVVPOS = 1
329*            LVINDX = 0
330*            LVFUNC=         STOP
331*            LVVARG=           R
332*            CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
333*            LVI     AAJ =           R
334*            IF (LVVAL.NE.-1) LVI     AAJ = LVVAL
335*            LVVTR = LVVAL
336*            LVVAL = -100
337*            IF (LVVTR.EQ.-1) GO TO          39
338*      C**** LVI     AAF             @            R
339*                    R = LVI     AAJ
340*            IF(STJ .LT. 0 .AND. KTMP .EQ. 889) GO TO 38
341*            ITEST=1
342*         32 CONTINUE
343*      C IF UNARY PLUS OR MINUS RETURN
344*            IF(STACK(JSTOP,1) .NE. 288 .AND. STACK(JSTOP,1) .NE. 110) GO TO 33
345*            JLAST=STACK(JSTOP,3)-1
346*            IF(STACK(JSTOP,1) .EQ. 288) JLAST=JLAST-1
347*      C     STRING+HOL,JLAST(-STRING,STRING @@TYPE(@@)
348*      C****      STRING           *          HOL
349*            LVVPOS =      JLAST
350*            LVVTYP =   3
351*            LVFUNC=         HOL
352*            LVVARG=      STRING
353*            CALL LVFIND(LV2       U,LV2       V,LV2       W,LV2       X)
354*            LVI     AAJ =      STRING
355*            IF (LVVAL.NE.-1) LVI     AAJ = LVVAL
356*            LVI     AAO = LVI     AAJ
357*      C**** LVI     AAL           -       STRING
```

```
358.            LVVAD=-1
359.            LVVTYP=-1
360.            LVVPOS=1
361.            LVFUNC=     STRING
362.            LVVARG=LV1     AAO
363.            CALL LVDLET
364.            LV1     AAO = LV1     AAJ
365.      C**** LV1     AAL       STRING          @@
366.            LVDEST=  0
367.            LV1     AAP = TYPE1
368.            LVTYPE(1) =  1
369.            LVVALS(1) = LV1     AAP
370.            LVDEST=  0
371.            LVVNVL =  1
372.            LVFUNC =      STRING
373.            LVVARG=LV1     AAO
374.            CALL LVNSRT
375.            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
376.            IF(LVVAL.LT.0) RETURN
377.      C     STRING+HOL,J(-STRING,STRING @@TYPE1@@)
378.      C****      STRING          +          HOL
379.            LVVPOS =          J
380.            LVVTYP =   3
381.            LVFUNC=         HOL
382.            LVVARG=      STRING
383.            CALL LVFIND(LV2       Y,LV2       Z,LV2       0,LV2       1)
384.            LV1     AAJ =      STRING
385.            IF (LVVAL.NE.-1) LV1     AAJ = LVVAL
386.            LV1     AAO = LV1     AAJ
387.      C**** LV1     AAL           -       STRING
388.            LVVAD=-1
389.            LVVTYP=-1
390.            LVVPOS=1
391.            LVFUNC=     STRING
392.            LVVARG=LV1     AAO
393.            CALL LVDLET
394.            LV1     AAO = LV1     AAJ
395.      C**** LV1     AAL       STRING          @@
396.            LVDEST=  0
397.            LV1     AAQ = TYPE1
398.            LVTYPE(1) =  1
399.            LVVALS(1) = LV1     AAQ
400.            LVDEST=  0
401.            LVVNVL =  1
402.            LVFUNC =      STRING
403.            LVVARG=LV1     AAO
404.            CALL LVNSRT
405.            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
406.            IF(LVVAL.LT.0) RETURN
407.            IF(ITEST .LT. 0) J=J-1
408.            RETURN
409.         33 CONTINUE
410.            IF(ITEST .LT. 0) J=J-1
411.            JSTACK=JSTACK+1
412.            STACK(JSTACK,1)=STOP
413.            STACK(JSTACK,2)=0
414.            STACK(JSTACK,3)=J
415.            STACK(JSTACK,4)=0
```

```
4160            NTMP=R
4170            CALL SLEVEL(SKIP)
4180            JSTACK=JSTACK-1
4190            R=NTMP
4200            JLAST=1
4210            IF(JSTOP .GT. 0) JLAST=STACK(JSTOP,3)
4220            NTMP=TYPE1
4230            JJ=JLAST
4240            IF(STACK(JSTACK,1) .EQ. 418 .AND. JLAST .GT. 1) JJ=JLAST-1
4250      C     STRING+HOL.JJ+STRING @TYPE1
4260      C**** STRING + HOL
4270            LVVPOS = JJ
4280            LVVTYP = 3
4290            LVFUNC= HOL
4300            LVVARG= STRING
4310            CALL LVFIND(LV2 2,LV2 3,LV2 4,LV2 5)
4320            LV1 AAJ = STRING
4330            IF (LVVAL.NE.-1) LV1 AAJ = LVVAL
4340      C**** LV1 AAF + STRING
4350            LVVTYP = 3
4360            LVVPOS = 1
4370            LVINDX = 0
4380            LVFUNC= STRING
4390            LVVARG= LV1 AAJ
4400            CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
4410            LV1 AAO = LV1 AAJ
4420            IF (LVVAL.NE.-1) LV1 AAO = LVVAL
4430      C**** LV1 AAL @ TYPE1
4440                TYPE1 = LV1 AAO
4450            IF(.NOT. FLAG .OR. NTMP .EQ. 3) GO TO 35
4460            IF((NTMP.EQ.0.OR.NTMP.EQ.2).AND.(TYPE1.EQ.0.OR.TYPE1.EQ.2))GOTO 35
4470            ERRFLG=.TRUE.
4480            CALL ERROR(78,J,KDM2)
4490         35 CONTINUE
4500            IF(TYPE1 .GT. 2 .OR. NTYPE .GT. 1) GO TO 38
4510            FUNC=FUNC1
4520            IF(TYPE1 .EQ. 1) FUNC=FUNC2
4530            IF(TYPE1 .EQ. 2) FUNC=FUNC3
4540      C     STRING+HOL.JLAST LEFT(.1 LPAR,.1 FUNC)
4550      C**** STRING + HOL
4560            LVVPOS = JLAST
4570            LVVTYP = 3
4580            LVFUNC= HOL
4590            LVVARG= STRING
4600            CALL LVFIND(LV2 6,LV2 7,LV2 8,LV2 9)
4610            LV1 AAO = STRING
4620            IF (LVVAL.NE.-1) LV1 AAO = LVVAL
4630            LV1 AAJ = LV1 AAO
4640            LV1 AAR = LEFT
4650      C**** LV1 AAO 1
4660            LVVTYP= 3
4670            LVVPOS= 1
4680      C**** LV1 AAF LV1 AAO LPAR
4690            LVDEST= 1
4700            LVTYPE(1) = 0
4710            LVVALS(1) = LPAR
4720            LVVNVL = 1
4730            LVFUNC = LV1 AAR
```

```
474•            LVVARG=LVI    AAJ
475•            CALL LVNSRT
476•            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
477•            IF(LVVAL.LT.0) RETURN
478•            LVI    AAJ = LVI    AAO
479•      C**** LVI    AAO           .          1
480•            LVVTYP=  3
481•            LVVPOS=        1
482•      C**** LVI    AAF  LVI    AAO         FUNC
483•            LVDEST=  1
484•            LVTYPE(1) =  0
485•            LVVALS(1) =       FUNC
486•            LVVNVL =  1
487•            LVFUNC = LVI    AAR
488•            LVVARG=LVI    AAJ
489•            CALL LVNSRT
490•            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
491•            IF(LVVAL.LT.0) RETURN
492•      C     STRING+HOL.J RIGHT RPAR
493•      C****     STRING          +          HOL
494•            LVVPOS =          J
495•            LVVTYP =  3
496•            LVFUNC=        HOL
497•            LVVARG=     STRING
498•            CALL LVFIND(LV2     AA,LV2     AB,LV2     AC,LV2     AD)
499•            LVI    AAO =     STRING
500•            IF (LVVAL.NE.-1) LVI    AAO = LVVAL
501•      C**** LVI    AAL       RIGHT         RPAR
502•            LVDEST=  0
503•            LVTYPE(1) =  0
504•            LVVALS(1) =       RPAR
505•            LVVNVL =  1
506•            LVFUNC =      RIGHT
507•            LVVARG=LVI    AAO
508•            CALL LVNSRT
509•            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
510•            IF(LVVAL.LT.0) RETURN
511•         38 IF(ITEST .LT. 0 .AND. STJ .LT. 0) J=J+1
512•            FLAG=.FALSE.
513•            RETURN
514•         39 FLAG=.FALSE.
515•            GO TO 11
516•      C  CHECK FOR CORRECTNESS OF SUBSCRIPTS
517•            NR=R
518•      C     R+STJ/11 OR
519•      C****          R          +          STJ
520•            LVVTYP = 3
521•            LVVPOS = 1
522•            LVINDX = 0
523•            LVFUNC=        STJ
524•            LVVARG=          R
525•            CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
526•            LVI    AAO =          R
527•            IF (LVVAL.NE.-1) LVI    AAO = LVVAL
528•            LVVTR = LVVAL
529•            LVVAL = -100
530•            IF (LVVTR.EQ.-1) GO TO          11
531•      C**** LVI    AAL          0           R
```

```
532*                    R = LVI     AAO
533*          NBETA=GETDIM(STR(J))
534*          IF(NR .EQ. 359 .AND. NBETA .NE. 4) GO TO 47
535*          ALPHA=GETTYP(STR(J))
536*          GAMMA=STR(J)/1000000
537*          IF(NTYPE .EQ. 3) GO TO 45
538*          IF(NR .EQ. 839 .AND. NBETA .EQ. 0) GO TO 45
539*          IF(NR .EQ. 359) GO TO 45
540*          IF(NR .EQ. 21 .AND. NBETA .EQ. 4) GO TO 45
541*          IF(NBETA .EQ. 0 .AND. NR .EQ. 935) GO TO 45
542*          IF(NTYPE .EQ. 2 .AND. NR .EQ. 935 .AND. STR(J-1) .NE. -7) GO TO 45
543*          IF(NTYPE .EQ. 2 .AND. NR .EQ. 359 .AND. NBETA .EQ. 0) GO TO 11
544*          CALL ERROR(79,J,KDM2)
545*          ERRFLG=.TRUE.
546*       45 CONTINUE
547*          IF(GAMMA .GE. 6 .AND. NBETA .EQ. 4) CALL ERROR(76,KDM1,KDM2)
548*          IF(ALPHA .EQ. 3) GO TO 46
549*          N1=ALPHA+1
550*          ERRFLG=.TRUE.
551*          CALL ERROR(80,TYPE(N1),J)
552*       46 IF(NBETA .EQ. 4) RETURN
553*          MARGS=MARGS+1
554*          LVI     AAO =      OPRAND
555*          LVDEST=  0
556*          LVI     AAR = MARGS
557*          LVTYPE(1) =  1
558*          LVVALS(1) = LVI     AAR
559*          LVDEST=  0
560*          LVVNVL =  1
561*          LVFUNC =       FUNC2
562*          LVVARG=LVI     AAO
563*          CALL LVNSRT
564*          IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
565*          IF(LVVAL.LT.0) RETURN
566*          LVDEST=  0
567*          LVI     AAJ = J
568*          LVTYPE(1) =  1
569*          LVVALS(1) = LVI     AAJ
570*          LVDEST=  0
571*          LVVNVL =  1
572*          LVFUNC =       FUNC3
573*          LVVARG=LVI     AAO
574*          CALL LVNSRT
575*          IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
576*          IF(LVVAL.LT.0) RETURN
577*          LVDEST=  0
578*          LVI     AAS = NDEPTH
579*          LVTYPE(1) =  1
580*          LVVALS(1) = LVI     AAS
581*          LVDEST=  0
582*          LVVNVL =  1
583*          LVFUNC =       LEVEL
584*          LVVARG=LVI     AAO
585*          CALL LVNSRT
586*          IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
587*          IF(LVVAL.LT.0) RETURN
588*          IVR=(MARGS+1)/2
589*          IF(IVR .GT. 50) GO TO 1610
```

```
590•            ICOL=18•MOD(MARGS-1,2)+9
591•            IVAL=MOD(STR(J),10000)
592•            IARGS(IVR)=BITPUT(IARGS(IVR),IVAL,ICOL)
593•            IF(NR .EQ. 839) GO TO 49
594•            IF(NTYPE .NE. 3) RETURN
595•      C  FLAG SUBSCRIPT IN I/O LIST
596•            IARGS(IVR)=BITPUT(IARGS(IVR),1,ICOL+3)
597•            RETURN
598•      C FLAG DO INDEX IN I/O LIST
599•         49 IARGS(IVR)=BITPUT(IARGS(IVR),2,ICOL+3)
600•            IF(NFLAG .LT. 1) RETURN
601•            IARGS(IVR)=BITPUT(IARGS(IVR),FL(NFLAG),ICOL+9)
602•            RETURN
603•         47 NR=R
604•      C    SUBSCRIPT DOES NOT BEGIN WITH CONSTANT, FORCE SEARCH FOR VARIABLE
605•            GO TO 11
606•      C  CHECK FOR PROPER NUMBER OF SUBSCRIPTS
607•         50 IF( BETA .EQ. 4 .OR. R .NE. 452) GO TO 52
608•            MARGS=MARGS+1
609•            LVI     AAO =      OPRAND
610•            LVDEST=  0
611•            LVI     AAT = MARGS
612•            LVTYPE(1) =  1
613•            LVVALS(1) = LVI     AAT
614•            LVDEST=  0
615•            LVVNVL =  1
616•            LVFUNC =       FUNC2
617•            LVVARG=LVI     AAO
618•            CALL LVNSRT
619•            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
620•            IF(LVVAL.LT.0) RETURN
621•            LVDEST=  0
622•            LVI     AAU = J-1
623•            LVTYPE(1) =  1
624•            LVVALS(1) = LVI     AAU
625•            LVDEST=  0
626•            LVVNVL =  1
627•            LVFUNC =       FUNC3
628•            LVVARG=LVI     AAO
629•            CALL LVNSRT
630•            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
631•            IF(LVVAL.LT.0) RETURN
632•            LVDEST=  0
633•            LVI     AAV = NDEPTH
634•            LVTYPE(1) =  1
635•            LVVALS(1) = LVI     AAV
636•            LVDEST=  0
637•            LVVNVL =  1
638•            LVFUNC =       LEVEL
639•            LVVARG=LVI     AAO
640•            CALL LVNSRT
641•            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
642•            IF(LVVAL.LT.0) RETURN
643•            IVR=(MARGS+1)/2
644•            IF(IVR .GT. 50) GO TO 1610
645•            ICOL=18•MOD(MARGS-1,2)+9
646•            IVAL=MOD(STR(J-1),10000)
647•            IARGS(IVR)=BITPUT(IARGS(IVR),IVAL,ICOL)
```

```
            IF(NOPAR .LE. 0) GO TO 52
            LVVPOS =          1
            LVVTYP =   3
            LVVPOS=-LVVPOS
            LVFUNC=      ACTION
            LVVARG=      OPRAND
            CALL LVFIND(LV2     AE,LV2     AF,LV2     AG,LV2     AH)
            LVI    AAO =     OPRAND
            IF (LVVAL.NE.-1) LVI    AAO = LVVAL
            LVVTR = LVVAL
            LVVAL = -100
            IF (LVVTR.EQ.-1) GO TO          52
                 MFUNC = LVI    AAO
            IARGS(IVR)=BITPUT(IARGS(IVR),MFUNC ,ICOL+3)
            IARGS(IVR)=BITPUT(IARGS(IVR),NARGS,ICOL+9)
C IF NO STRING LEFT, RETURN IF CONSTANT,VARIABLE OR I/O LIST
   52 IF(STJ .LT. 0 .AND. (BETA .EQ. 0 .OR. BETA .EQ. 4
     $ .OR. NTYPE .EQ. 3)) RETURN
            IF(BETA .GT. NARRAY) GO TO 55
            ERRFLG=.TRUE.
            CALL ERROR(81,J)
   55 IF(R .EQ. 452) NARRAY=0
            IF(R .EQ. 318) NARRAY=1
            IF(R .EQ.  60) NARRAY=2
            IF(R .EQ. 103) NARRAY=3
            IF(STJ .LT. 0) GO TO 58
C           R+STJ @R//56
C++++            R            +          STJ
            LVVTYP = 3
            LVVPOS = 1
            LVINDX = 0
            LVFUNC=          STJ
            LVVARG=           R
            CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
            LVI    AAO =          R
            IF (LVVAL.NE.-1) LVI    AAO = LVVAL
C++++ LVI    AAL           @           R
                 R = LVI    AAO
            LVVTR = LVVAL
            LVVAL = -100
            IF (LVVTR.NE.-1) GO TO          56
   58 IF(NTYPE .EQ. 3 .AND. NARRAY .EQ. 0) GO TO 57
            IF(BETA .GE. 1 .AND. BETA .LE. 3 .AND. NOPAR .EQ. 0)
     $ CALL ERROR(82,J,KOM2)
   57 NARRAY=-1
            GO TO 11
   56 IF(NTYPE .EQ. 3 .AND. NARRAY .EQ. 0) RETURN
            IF(STJ .EQ. RPAR .AND. NARRAY .LT.  BETA .AND. J .EQ. MAXJ)
     $ CALL ERROR(82,J,KOM2)
            IF(STJ .EQ. RPAR) NARRAY=-1
            RETURN
C  RESET TYPE OF STATEMENT IN ANTICIPATION OF SEARCH FOR BOOLEAN PRIMARY
   60 CONTINUE
            NOTFLG=.FALSE.
            IF(STR(J-1) .EQ. -17) NOTFLG=.TRUE.
            TYPE1=-1
C           STRING+HOL.J(-STRING,STRING @@TYPE1@@)
C++++       STRING            +          HOL
```

```
706•            LVVPOS =           J
707•            LVVTYP =   3
708•            LVFUNC=         HOL
709•            LVVARG=      STRING
710•            CALL LVFIND(LV2      AI,LV2      AJ,LV2      AK,LV2      AL)
711•            LVI     AAO =      STRING
712•            IF (LVVAL.NE.-1) LVI     AAO = LVVAL
713•            LVI     AA@ = LVI     AAO
714•      C**** LVI     AAO            -       STRING
715•            LVVAD=-1
716•            LVVTYP=-1
717•            LVVPOS=1
718•            LVFUNC=     STRING
719•            LVVARG=LVI     AA@
720•            CALL LVDLET
721•            LVI     AA@ = LVI     AAO
722•      C**** LVI     AAO       STRING            @@
723•            LVDEST=  0
724•            LVI     AAX = TYPE1
725•            LVTYPE(1) =  1
726•            LVVALS(1) = LVI     AAX
727•            LVDEST=  0
728•            LVVNVL =  1
729•            LVFUNC =      STRING
730•            LVVARG=LVI     AA@
731•            CALL LVNSRT
732•            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
733•            IF(LVVAL.LT.0) RETURN
734•            IF(STJ .NE. OPRAND) GO TO 65
735•            ALPHA=GETTYP(STR(J))
736•            BETA=GETDIM(STR(J))
737•            IF(ALPHA .NE. 4) GO TO 11
738•         65 CONTINUE
739•      C     R+STJ/11 @R
740•      C****           R            +          STJ
741•            LVVTYP = 3
742•            LVVPOS = 1
743•            LVINDX = 0
744•            LVFUNC=         STJ
745•            LVVARG=           R
746•            CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
747•            LVI     AAO =           R
748•            IF (LVVAL.NE.-1) LVI     AAO = LVVAL
749•            LVVTR = LVVAL
750•            LVVAL = -100
751•            IF (LVVTR.EQ.-1) GO TO          11
752•      C**** LVI     AAL             @            R
753•                    R = LVI     AAO
754•            RETURN
755•      C  IF BOOLEAN PRIMARY IS AN ARETHMETIC COMPARE CONTINUE PARSING PRIMARY
756•         70 IF(STJ .LT. 0) RETURN
757•            IF(TYPE1 .EQ. 4) GO TO 75
758•      C     R+STJ @R/11
759•      C****           R            +          STJ
760•            LVVTYP = 3
761•            LVVPOS = 1
762•            LVINDX = 0
763•            LVFUNC=         STJ
```

```
      LVVARG=           R
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LVI    AAO =           R
      IF (LVVAL.NE.-1) LVI    AAO = LVVAL
C**** LVI    AAL             B           R
                R = LVI    AAO
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO          11
C  RELATIONAL OPERATOR FOUND
      IF(NDEP .EQ. 0) RETURN
      LVVPOS=-LVVPOS
      LVVTYP=  3
      LVVPOS=          1
      LVDEST=  2
      LVI    AAO = 1
      LVTYPE(1) =  1
      LVVALS(1) = LVI    AAO
      LVDEST=  2
      LVVNVL =  1
      LVFUNC =      FUNC1
      LVVARG=    OPRAND
      CALL LVNSRT
      IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
      IF(LVVAL.LT.0) RETURN
      RETURN
C  IF BOOLEAN VARIABLE OR CONSTANT, SET STATE TO STOP
   75 R=STOP
      JSTACK=JSTACK+1
      STACK(JSTACK,1)=R
      STACK(JSTACK,2)=0
      STACK(JSTACK,3)=J
      STACK(JSTACK,4)=0
      GO TO 11
C  COMPARE TYPES ON BOTH SIDES OF RELATIONAL EXPRESSION
   80 IF(TYPE1 .EQ. 0 .OR. TYPE1 .EQ. 2 .OR. TYPE1 .EQ. 3) GO TO 85
      ERRFLG=.TRUE.
      CALL ERROR(83,J,KDM2)
      TYPE1=-1
C     STRING+HOL,J(-STRING,STRING @@TYPE1@@)
C****      STRING                    HOL
      LVVPOS =          J
      LVVTYP =  3
      LVFUNC=        HOL
      LVVARG=     STRING
      CALL LVFIND(LV2     AM,LV2     AN,LV2     AO,LV2     AP)
      LVI    AA@ =     STRING
      IF (LVVAL.NE.-1) LVI    AA@ = LVVAL
      LVI    AAY = LVI    AA@
C**** LVI    AAO           -       STRING
      LVVAD=-1
      LVVTYP=-1
      LVVPOS=1
      LVFUNC=    STRING
      LVVARG=LVI    AAY
      CALL LVDLET
      LVI    AAY = LVI    AA@
C**** LVI    AAO      STRING          @@
```

```
822•            LVDEST= 0
823•            LVI    AAZ = TYPE1
824•            LVTYPE(1) = 1
825•            LVVALS(1) = LVI    AAZ
826•            LVDEST= 0
827•            LVVNVL = 1
828•            LVFUNC =     STRING
829•            LVVARG=LVI    AAY
830•            CALL LVNSRT
831•            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
832•            IF(LVVAL.LT.0) RETURN
833•         85 TYPE2=TYPE1
834•            GO TO 11
835•      C  BOOLEAN PRIMARY RECOGNIZED-SET TYPE TO BOOLEAN AND CONTINUE PARSE
836•         90 IF(TYPE1 .EQ. TYPE2 .OR. TYPE1+TYPE2 .EQ. 2 .OR.
837•           + TYPE2 .LT. 0) GO TO 95
838•            N1=TYPE1+1
839•            N2=TYPE2+1
840•            CALL ERROR(77,TYPE(N1),TYPE(N2))
841•            ERRFLG=.TRUE.
842•         95 TYPE1=9
843•            TYPE2=-1
844•            IF(STJ .LT. 0) RETURN
845•      C     STRING+HOL,J1-STRING,STRING @@TYPE1@@)
846•      C****      STRING          +          HOL
847•            LVVPOS =          J
848•            LVVTYP =   3
849•            LVFUNC=        HOL
850•            LVVARG=     STRING
851•            CALL LVFIND(LV2    AQ,LV2    AR,LV2    AS,LV2    AT)
852•            LVI    AA# =     STRING
853•            IF (LVVAL.NE.-1) LVI    AA# = LVVAL
854•            LVI    AAY = LVI    AA#
855•      C**** LVI    AAQ          -      STRING
856•            LVVAD=-1
857•            LVVTYP=-1
858•            LVVPOS=1
859•            LVFUNC=    STRING
860•            LVVARG=LVI    AAY
861•            CALL LVDLET
862•            LVI    AAY = LVI    AA#
863•      C**** LVI    AAQ      STRING          @@
864•            LVDEST= 0
865•            LVI    AAQ = TYPE1
866•            LVTYPE(1) = 1
867•            LVVALS(1) = LVI    AAQ
868•            LVDEST= 0
869•            LVVNVL = 1
870•            LVFUNC =     STRING
871•            LVVARG=LVI    AAY
872•            CALL LVNSRT
873•            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
874•            IF(LVVAL.LT.0) RETURN
875•            GO TO 11
876•      C  PARSE REACHED BLIND ALLEY-MUST BACK UP AND REMOVE PARENTHESES CREATED
877•        999 JM=STACK(JSTACK,3)
878•            K=JM
879•            DO 996 KK=JM,J
```

```
C      STRING+HOL.K/995+STRING.I/996 @TYPE)
C****      STRING              +          HOL
           LVVPOS =            K
           LVVTYP =   3
           LVFUNC=          HOL
           LVVARG=      STRING
           CALL LVFIND(LV2      AU,LV2      AV,LV2      AW,LV2      AX)
           LV1     AA# =      STRING
           IF (LVVAL.NE.-1) LV1     AA# = LVVAL
           LVVTR = LVVAL
           LVVAL = -100
           IF (LVVTR.EQ.-1) GO TO          995
C**** LV1     AAL              +       STRING
           LVVPOS =            1
           LVVTYP =   3
           LVFUNC=      STRING
           LVVARG= LV1     AA#
           CALL LVFIND(LV2      AY,LV2      AZ,LV2      A0,LV2      A1)
           LV1     AAY = LV1     AA#
           IF (LVVAL.NE.-1) LV1     AAY = LVVAL
           LVVTR = LVVAL
           LVVAL = -100
           IF (LVVTR.EQ.-1) GO TO          996
C**** LV1     AA0              @         TYPE)
                TYPE) = LV1     AAY
           GO TO 995
     996 K=K-1
     995 CONTINUE
           DO 998 I=JM,J
C      STRING+HOL.I(-LEFT,-RIGHT)
C****      STRING              +          HOL
           LVVPOS =            1
           LVVTYP =   3
           LVFUNC=          HOL
           LVVARG=      STRING
           CALL LVFIND(LV2      A2,LV2      A3,LV2      A4,LV2      A5)
           LV1     AAY =      STRING
           IF (LVVAL.NE.-1) LV1     AAY = LVVAL
           LV1     AA# = LV1     AAY
C**** LV1     AAL              -         LEFT
           LVVAD=-1
           LVVTYP=-1
           LVVPOS=1
           LVFUNC=       LEFT
           LVVARG=LV1     AA#
           CALL LVDLET
           LV1     AA# = LV1     AAY
C**** LV1     AAL              -         RIGHT
           LVVAD=-1
           LVVTYP=-1
           LVVPOS=1
           LVFUNC=      RIGHT
           LVVARG=LV1     AA#
           CALL LVDLET
     998 CONTINUE
     980 CONTINUE
           LVVPOS =            1
           LVVTYP =   3
```

```
      LVVPOS=-LVVPOS
      LVFUNC=      FUNC3
      LVVARG=     OPRAND
      CALL LVFIND(LV2     A6,LV2     A7,LV2     A8,LV2     A9)
      LV1     AAY =      OPRAND
      IF (LVVAL.NE.-1) LV1     AAY = LVVAL
           JN = LV1     AAY
      IF(JN .LT. JM) GO TO 985
      LV1     AAY =      OPRAND
      LVVPOS =          1
      LVVTYP =   3
      LVVPOS=-LVVPOS
      LVFUNC=      FUNC2
      LVVARG= LV1     AAY
      LVVAD=-1
      CALL LVDLET
      LVVPOS =          1
      LVVTYP =   3
      LVVPOS=-LVVPOS
      LVFUNC=      FUNC3
      LVVARG= LV1     AAY
      LVVAD=-1
      CALL LVDLET
      LVVPOS =          1
      LVVTYP =   3
      LVVPOS=-LVVPOS
      LVFUNC=      LEVEL
      LVVARG= LV1     AAY
      LVVAD=-1
      CALL LVDLET
      GO TO 980
  985 CONTINUE
      LVVPOS =          1
      LVVTYP =   3
      LVVPOS=-LVVPOS
      LVFUNC=      FUNC2
      LVVARG=     OPRAND
      CALL LVFIND(LV2     BA,LV2     BB,LV2     BC,LV2     BD)
      LV1     AAY =      OPRAND
      IF (LVVAL.NE.-1) LV1     AAY = LVVAL
           NARGS = LV1     AAY
      LVVPOS =          1
      LVVTYP =   3
      LVVPOS=-LVVPOS
      LVFUNC=      LEVEL
      LVVARG=     OPRAND
      CALL LVFIND(LV2     BE,LV2     BF,LV2     BG,LV2     BH)
      LV1     AAY =      OPRAND
      IF (LVVAL.NE.-1) LV1     AAY = LVVAL
           NDEPTH = LV1     AAY
      RETURN

C     RECOGNIZED FUNCTION-PREPARE TO SET TYPE OF ARGUMENTS FOR THE @NDEPTH@
C               FUNCTION IN THIS STMT
 1000 CONTINUE
      NDEPTH=NDEPTH+1
      NDEP=NDEP+1
      NARGS=0
```

```
 996.     C      R=STJ/11 @R
 997.     C****          R          +          STJ
 998.            LVVTYP = 3
 999.            LVVPOS = 1
1000.            LVINDX = 0
1001.            LVFUNC=          STJ
1002.            LVVARG=           R
1003.            CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
1004.            LVI     AAY =           R
1005.            IF (LVVAL.NE.-1) LVI     AAY = LVVAL
1006.            LVVTR = LVVAL
1007.            LVVAL = -100
1008.            IF (LVVTR.EQ.-1) GO TO          11
1009.     C**** LVI     AAO           @           R
1010.                    R = LVI     AAY
1011.            NARGS=1
1012.     C      OPRAND(OPRAND @@TYPE1@@,STRING@@NARGS@@,ACTION @@NDEPTH@@)
1013.            LVI     AAY =      OPRAND
1014.     C**** LVI     AAO       OPRAND           @@
1015.            LVDEST=  0
1016.            LVI     AA@ = TYPE1
1017.            LVTYPE(1) =  1
1018.            LVVALS(1) = LVI     AA@
1019.            LVDEST=  0
1020.            LVVNVL =  1
1021.            LVFUNC =      OPRAND
1022.            LVVARG=LVI     AAY
1023.            CALL LVNSRT
1024.            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
1025.            IF(LVVAL.LT.0) RETURN
1026.     C**** LVI     AAO       STRING           @@
1027.            LVDEST=  0
1028.            LVI     AA1 = NARGS
1029.            LVTYPE(1) =  1
1030.            LVVALS(1) = LVI     AA1
1031.            LVDEST=  0
1032.            LVVNVL =  1
1033.            LVFUNC =      STRING
1034.            LVVARG=LVI     AAY
1035.            CALL LVNSRT
1036.            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
1037.            IF(LVVAL.LT.0) RETURN
1038.     C**** LVI     AAO       ACTION           @@
1039.            LVDEST=  0
1040.            LVI     AA2 = NDEPTH
1041.            LVTYPE(1) =  1
1042.            LVVALS(1) = LVI     AA2
1043.            LVDEST=  0
1044.            LVVNVL =  1
1045.            LVFUNC =      ACTION
1046.            LVVARG=LVI     AAY
1047.            CALL LVNSRT
1048.            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
1049.            IF(LVVAL.LT.0) RETURN
1050.     C      OPRAND FUNC1 @@J+1@@
1051.     C****      OPRAND       FUNC1           @@
1052.            LVDEST=  0
1053.            LVI     AAY = 0
```

```
1054•            LVTYPE(I) =  1
1055•            LVVALS(I) = LVI     AAY
1056•            LVDEST=  0
1057•            LVVNVL =  1
1058•            LVFUNC =       FUNCI
1059•            LVVARG=     OPRAND
1060•            CALL LVNSRT
1061•            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
1062•            IF(LVVAL.LT.0) RETURN
1063•            TYPEI=-1
1064•      C     NOPAR=NOPAR+1
1065•            NOPAR=NOPAR+1
1066•            RETURN
1067•
1068•      C KEEP TRACK OF THE NUMBER AND TYPES OF ARGUMENTS IN FUNCTION CALLS
1069•      C   MUST USE STACK FOR POSSIBLE RECURSIVE FUNCTION USE
1070•       1100 CONTINUE
1071•            LVVTYP = 3
1072•            LVVPOS = 1
1073•            LVINDX = 0
1074•            LVFUNC=          STJ
1075•            LVVARG=            R
1076•            CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
1077•            LVI     AA3 =            R
1078•            IF (LVVAL.NE.-1) LVI     AA3 = LVVAL
1079•            LVVTR = LVVAL
1080•            LVVAL = -100
1081•            IF (LVVTR.EQ.-1) GO TO           11
1082•                     R = LVI     AA3
1083•            LVI     AA3 =      OPRAND
1084•            LVVPOS =            1
1085•            LVVTYP =   3
1086•            LVVPOS=-LVVPOS
1087•            LVFUNC=      STRING
1088•            LVVARG= LVI     AA3
1089•            CALL LVFIND(LV2      BI,LV2      BJ,LV2      BK,LV2      BL)
1090•            LVI     AA4 = LVI     AA3
1091•            IF (LVVAL.NE.-1) LVI     AA4 = LVVAL
1092•            LVVTR = LVVAL
1093•            LVVAL = -100
1094•            IF (LVVTR.EQ.-1) GO TO         1103
1095•      C**** LVI     AAU             @        NARGS
1096•                NARGS = LVI     AA4
1097•      C**** LVI     AAT             +       STRING
1098•            LVVPOS =           1
1099•            LVVTYP =   3
1100•            LVVPOS=-LVVPOS
1101•            LVFUNC=      STRING
1102•            LVVARG= LVI     AA3
1103•            LVVAD=-1
1104•            CALL LVDLET
1105•      C**** LVI     AAT             +       ACTION
1106•            LVVPOS =           1
1107•            LVVTYP =   3
1108•            LVVPOS=-LVVPOS
1109•            LVFUNC=      ACTION
1110•            LVVARG= LVI     AA3
1111•            CALL LVFIND(LV2      BM,LV2      BN,LV2      BO,LV2      BP)
```

```
1112*           LVI     AA4 = LVI     AA3
1113*           IF (LVVAL.NE.-1) LVI     AA4 = LVVAL
1114*     C**** LVI     AAU             @*      NDEPTH
1115*                MFUNC = LVI     AA4
1116*           LVVPOS =           1
1117*           LVVTYP =   3
1118*           LVVPOS=-LVVPOS
1119*           LVFUNC=      FUNC)
1120*           LVVARG=     OPRAND
1121*           CALL LVFIND(LV2     BQ,LV2     BR,LV2     BS,LV2     BT)
1122*           LVI     AA3 =     OPRAND
1123*           IF (LVVAL.NE.-1) LVI     AA3 = LVVAL
1124*                IEXP = LVI     AA3
1125*     C1103 R=STJ/11 BR
1126*      1103 CONTINUE
1127*     C  STORE ARGUMENT TYPES
1128*           IF(NDEPTH .GT. 5) CALL ERROR(85)
1129*           IF(NDEPTH .GT. 5) GO TO 1130
1130*           IF(NARGS .LE. 63) GO TO 1104
1131*           ERRFLG=.TRUE.
1132*           CALL ERROR(84,NDEPTH,KOM2)
1133*           GO TO 11
1134*      1104 CONTINUE
1135*           MM=(7+NARGS)/4
1136*           ITEMP=NARGS-4*(MM-2)
1137*           ICOL=9*ITEMP-6
1138*           NARY(MFUNC,MM)=BITPUT(NARY(MFUNC,MM),MOD((TYPEI+1),6),ICOL)
1139*           IF(STR(J-2) .NE. -6 .AND. STR(J-2) .NE. -4) GO TO 1130
1140*           NDIM=GETDIM(STR(J-1))
1141*           IF(NDIM .GE. 4) GO TO 1130
1142*     C  STORE DIMENSIONALITY OF ARGUMENTS
1143*           NARY(MFUNC ,MM)=BITPUT(NARY(MFUNC ,MM),NDIM,ICOL+3)
1144*      1130 CONTINUE
1145*           NARY(MFUNC,MM)=BITPUT(NARY(MFUNC,MM),IEXP,ICOL+6)
1146*           IF(STJ .EQ. COMMA) GO TO 1105
1147*           NARY(MFUNC,1)=NARGS
1148*           LVI     AA3 =     OPRAND
1149*           LVVPOS =           1
1150*           LVVTYP =   3
1151*           LVVPOS=-LVVPOS
1152*           LVFUNC=     ACTION
1153*           LVVARG= LVI     AA3
1154*           LVVAD=-1
1155*           CALL LVDLET
1156*           LVVPOS =           1
1157*           LVVTYP =   3
1158*           LVVPOS=-LVVPOS
1159*           LVFUNC=     OPRAND
1160*           LVVARG= LVI     AA3
1161*           CALL LVFIND(LV2     BU,LV2     BV,LV2     BW,LV2     BX)
1162*           LVI     AA4 = LVI     AA3
1163*           IF (LVVAL.NE.-1) LVI     AA4 = LVVAL
1164*           LVVTR = LVVAL
1165*           LVVAL = -100
1166*           IF (LVVTR.EQ.-1) GO TO       1135
1167*                TYPE) = LVI     AA4
1168*           LVI     AA3 =     OPRAND
1169*           LVVPOS =           1
```

```
1170•            LVVTYP =   3
1171•            LVVPOS=-LVVPOS
1172•            LVFUNC=       FUNC1
1173•            LVVARG= LV1     AA3
1174•            LVVAD=-1
1175•            CALL LVDLET
1176•            LVVPOS =            1
1177•            LVVTYP =   3
1178•            LVVPOS=-LVVPOS
1179•            LVFUNC=      OPRAND
1180•            LVVARG= LV1     AA3
1181•            LVVAD=-1
1182•            CALL LVDLET
1183•       1135 NOPAR=NOPAR-1
1184•            NDEP=NDEP-1
1185•            RETURN
1186•        1105 TYPE1=-1
1187•       C     STRING+HOL,J(-STRING,STRING @@TYPE1@@)
1188•       C****      STRING            +           HOL
1189•            LVVPOS =          J
1190•            LVVTYP =   3
1191•            LVFUNC=         HOL
1192•            LVVARG=      STRING
1193•            CALL LVFIND(LV2      BY,LV2      BZ,LV2      BQ,LV2      B1)
1194•            LV1     AA3 =      STRING
1195•            IF (LVVAL.NE.-1) LV1     AA3 = LVVAL
1196•            LV1     AA4 = LV1     AA3
1197•       C**** LV1     AAU             -       STRING
1198•            LVVAD=-1
1199•            LVVTYP=-1
1200•            LVVPOS=1
1201•            LVFUNC=    STRING
1202•            LVVARG=LV1     AA4
1203•            CALL LVDLET
1204•            LV1     AA4 = LV1     AA3
1205•       C**** LV1     AAU       STRING           @@
1206•            LVDEST=  0
1207•            LV1     AA5 = TYPE1
1208•            LVTYPE(1) =  1
1209•            LVVALS(1) = LV1     AA5
1210•            LVDEST=  0
1211•            LVVNVL =  1
1212•            LVFUNC =      STRING
1213•            LVVARG=LV1     AA4
1214•            CALL LVNSRT
1215•            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
1216•            IF(LVVAL.LT.0) RETURN
1217•            NARGS=NARGS+1
1218•       C**** LV1     AAT       STRING           @@
1219•            LVDEST=  0
1220•            LV1     AA3 = NARGS
1221•            LVTYPE(1) =  1
1222•            LVVALS(1) = LV1     AA3
1223•            LVDEST=  0
1224•            LVVNVL =  1
1225•            LVFUNC =      STRING
1226•            LVVARG=     OPRAND
1227•            CALL LVNSRT
```

```
1228*           IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
1229*           IF(LVVAL.LT.0) RETURN
1230*     C**** FUNC1                          1
1231*           LVVPOS=-LVVPOS
1232*           LVVTYP=  3
1233*           LVVPOS=          1
1234*     C**** LV1    AA1       FUNC1          @@
1235*           LVDEST=  2
1236*           LV1    AA4 = 0
1237*           LVTYPE(1) =  1
1238*           LVVALS(1) = LV1    AA4
1239*           LVDEST=  2
1240*           LVVNVL =  1
1241*           LVFUNC =      FUNC1
1242*           LVVARG=    OPRAND
1243*           CALL LVNSRT
1244*           IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
1245*           IF(LVVAL.LT.0) RETURN
1246*           RETURN
1247*     C  SAVE TYPE OF STAEMENT WHILE PARSING EXPONENT
1248*      1200 CONTINUE
1249*     C     STRING=HOL.J(=STRING,STRING @@TYPE1@@)
1250*     C**** STRING           +          HOL
1251*           LVVPOS =          J
1252*           LVVTYP =   3
1253*           LVFUNC=         HOL
1254*           LVVARG=      STRING
1255*           CALL LVFIND(LV2    B2,LV2    B3,LV2    B4,LV2    B5)
1256*           LV1    AA6 =     STRING
1257*           IF (LVVAL.NE.-1) LV1    AA6 = LVVAL
1258*           LV1    AA7 = LV1    AA6
1259*     C**** LV1    AAX          =      STRING
1260*           LVVAD=-1
1261*           LVVTYP=-1
1262*           LVVPOS=1
1263*           LVFUNC=    STRING
1264*           LVVARG=LV1    AA7
1265*           CALL LVDLET
1266*           LV1    AA7 = LV1    AA6
1267*     C**** LV1    AAX      STRING          @@
1268*           LVDEST=  0
1269*           LV1    AA8 = TYPE1
1270*           LVTYPE(1) =  1
1271*           LVVALS(1) = LV1    AA8
1272*           LVDEST=  0
1273*           LVVNVL =  1
1274*           LVFUNC =     STRING
1275*           LVVARG=LV1    AA7
1276*           CALL LVNSRT
1277*           IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
1278*           IF(LVVAL.LT.0) RETURN
1279*           TYPE1=-1
1280*     C     R=STJ/11 @R
1281*     C**** R           +          STJ
1282*           LVVTYP = 3
1283*           LVVPOS = 1
1284*           LVINDX = 0
1285*           LVFUNC=        STJ
```

```
1286•            LVVARG=           R
1287•            CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
1288•            LVI     AA6 =             R
1289•            IF (LVVAL.NE.-1) LVI     AA6 = LVVAL
1290•            LVVTR = LVVAL
1291•            LVVAL = -100
1292•            IF (LVVTR.EQ.-1) GO TO          11
1293•       C••••• LVI     AAT            R            R
1294•                     R = LVI     AA6
1295•       C     COMPLETE
1296•            RETURN
1297•        1300 CONTINUE
1298•            IF(STJ .LT. 0) RETURN
1299•            IF(STJ .NE. AND .AND. STJ .NE. OR .AND. STJ .NE. NOT) GO TO 11
1300•            LVVTYP = 3
1301•            LVVPOS = 1
1302•            LVINDX = 0
1303•            LVFUNC=         STJ
1304•            LVVARG=           R
1305•            CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
1306•            LVI     AA6 =           R
1307•            IF (LVVAL.NE.-1) LVI     AA6 = LVVAL
1308•            LVVTR = LVVAL
1309•            LVVAL = -100
1310•            IF (LVVTR.EQ.-1) GO TO          11
1311•                     R = LVI     AA6
1312•            IF(NDEP .EQ. 0) RETURN
1313•       C  LOGICAL OPERATOR FOUND
1314•            LVVPOS=-LVVPOS
1315•            LVVTYP=  3
1316•            LVVPOS=          1
1317•            LVDEST=  2
1318•            LVI     AA6 = 1
1319•            LVTYPE(1) =  1
1320•            LVVALS(1) = LVI     AA6
1321•            LVDEST=  2
1322•            LVVNVL =  1
1323•            LVFUNC =       FUNC1
1324•            LVVARG=     OPRAND
1325•            CALL LVNSRT
1326•            IF(LVVAL.LT.0) CALL LVEXIT(LVVAL)
1327•            IF(LVVAL.LT.0) RETURN
1328•            RETURN
1329•        1900 CONTINUE
1330•            LVVTYP = 3
1331•            LVVPOS = 1
1332•            LVINDX = 0
1333•            LVFUNC=         STJ
1334•            LVVARG=           R
1335•            CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
1336•            LVI     AA7 =           R
1337•            IF (LVVAL.NE.-1) LVI     AA7 = LVVAL
1338•            LVVTR = LVVAL
1339•            LVVAL = -100
1340•            IF (LVVTR.EQ.-1) GO TO          11
1341•                     R = LVI     AA7
1342•       C  LEFT PAREN FOUND IN I/O LIST
1343•            NFLAG=NFLAG+1
```

```
1344•            FL(NFLAG)=MARGS
1345•            RETURN
1346•       1500 CONTINUE
1347•            LVVTYP = 3
1348•            LVVPOS = 1
1349•            LVINDX = 0
1350•            LVFUNC=          STJ
1351•            LVVARG=            R
1352•            CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
1353•            LVI     AA7 =            R
1354•            IF (LVVAL.NE.-1) LVI     AA7 = LVVAL
1355•            LVVTR = LVVAL
1356•            LVVAL = -100
1357•            IF (LVVTR.EQ.-1) GO TO          11
1358•                     R = LVI     AA7
1359•            IF(STJ .EQ. COMMA) RETURN
1360•       C  RIGHT PAREN FOUND IN I/O LIST
1361•            NFLAG=NFLAG-1
1362•            RETURN
1363•       1600 CONTINUE
1364•            TYPE1=4
1365•            RETURN
1366•       1610 CALL ERROR(95,IDM1,IDM2)
1367•            STOP
1368•      25000 CONTINUE
1369•            LV2A=0
1370•            LV2B=0
1371•            LV2C=0
1372•            LV2D=0
1373•            LV2E=0
1374•            LV2F=0
1375•            LV2G=0
1376•            LV2H=0
1377•            LV2I=0
1378•            LV2J=0
1379•            LV2K=0
1380•            LV2L=0
1381•            LV2M=0
1382•            LV2N=0
1383•            LV2O=0
1384•            LV2P=0
1385•            LV2Q=0
1386•            LV2R=0
1387•            LV2S=0
1388•            LV2T=0
1389•            LV2U=0
1390•            LV2V=0
1391•            LV2W=0
1392•            LV2X=0
1393•            LV2Y=0
1394•            LV2Z=0
1395•            LV2D=0
1396•            LV21=0
1397•            LV22=0
1398•            LV23=0
1399•            LV24=0
1400•            LV25=0
1401•            LV26=0
```

```
1402.          LV27=0
1403.          LV28=0
1404.          LV29=0
1405.          LV2AA=0
1406.          LV2AB=0
1407.          LV2AC=0
1408.          LV2AD=0
1409.          LV2AE=0
1410.          LV2AF=0
1411.          LV2AG=0
1412.          LV2AH=0
1413.          LV2AI=0
1414.          LV2AJ=0
1415.          LV2AK=0
1416.          LV2AL=0
1417.          LV2AM=0
1418.          LV2AN=0
1419.          LV2AO=0
1420.          LV2AP=0
1421.          LV2AQ=0
1422.          LV2AR=0
1423.          LV2AS=0
1424.          LV2AT=0
1425.          LV2AU=0
1426.          LV2AV=0
1427.          LV2AW=0
1428.          LV2AX=0
1429.          LV2AY=0
1430.          LV2AZ=0
1431.          LV2A0=0
1432.          LV2A1=0
1433.          LV2A2=0
1434.          LV2A3=0
1435.          LV2A4=0
1436.          LV2A5=0
1437.          LV2A6=0
1438.          LV2A7=0
1439.          LV2A8=0
1440.          LV2A9=0
1441.          LV2BA=0
1442.          LV2BB=0
1443.          LV2BC=0
1444.          LV2BD=0
1445.          LV2BE=0
1446.          LV2BF=0
1447.          LV2BG=0
1448.          LV2BH=0
1449.          LV2BI=0
1450.          LV2BJ=0
1451.          LV2BK=0
1452.          LV2BL=0
1453.          LV2BM=0
1454.          LV2BN=0
1455.          LV2BO=0
1456.          LV2BP=0
1457.          LV2BQ=0
1458.          LV2BR=0
1459.          LV2BS=0
1460.          LV2BT=0
1461.          LV2BU=0
1462.          LV2BV=0
1463.          LV2BW=0
1464.          LV2BX=0
1465.          LV2BY=0
1466.          LV2BZ=0
1467.          LV2B1=0
1468.          LV2B2=0
1469.          LV2B3=0
1470.          LV2B4=0
1471.          LV2B5=0
1472.          GO TO 25001
1473.          END
```

```
      SUBROUTINE SEPAR                                                  SEPAR  2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      COMMON/FORMAT/IDESST,IDESND,IGPST,IGPND,IGRP,SEPST,SEPND,
     1 DIR,ICOM,ISEP                                                    SEPAR  5
      INTEGER A,SEPST,SEPND,DIR,BLANK,SLASH,COMMA                       SEPAR  6
      DATA BLANK/1H /,SLASH/1H//,COMMA/1H,/                             SEPAR  7
      ICOM=0                                                            SEPAR  8
      ISLASH=0                                                          SEPAR  9
      DO 20 I=1,N                                                       SEPAR 10
      JJ=SEPST+DIR*(I-1)
      IF(A(JJ) .EQ. BLANK) GO TO 20                                     SEPAR 12
      IF(A(JJ) .EQ. SLASH) GO TO 5                                      SEPAR 13
      IF(A(JJ) .EQ. COMMA) GO TO 10                                     SEPAR 14
      GO TO 30                                                          SEPAR 15
    5 CONTINUE                                                          SEPAR 16
      ISLASH=1                                                          SEPAR 17
      IF(ICOM .EQ. 1) GO TO 40                                          SEPAR 18
      GO TO 20                                                          SEPAR 19
   10 IF(ISLASH .EQ. 1 .OR. ICOM .EQ. 1) GO TO 40                       SEPAR 20
      ICOM=1                                                            SEPAR 21
   20 CONTINUE                                                          SEPAR 22
      GO TO 40                                                          SEPAR 23
   30 IF(ISLASH .EQ. 0 .AND. ICOM .EQ. 0) GO TO 35                      SEPAR 24
      ISEP=1                                                            SEPAR 25
      SEPND=JJ-DIR
      RETURN                                                            SEPAR 27
   35 CONTINUE                                                          SEPAR 28
      ISEP=0                                                            SEPAR 29
      SEPND=JJ                                                          SEPAR 30
      RETURN                                                            SEPAR 31
   40 ISEP=-1                                                           SEPAR 32
      RETURN                                                            SEPAR 33
      END                                                               SEPAR 34
```

```
      SUBROUTINE SIMP                                                   SIMP  2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH  2
     • JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH  4
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      DIMENSION IALPH1(7),IALPH2(9),IALPH3(5),IALPH4(10)                SIMP  5
      DATA ((IALPH1(I),I=1,7)/1HR,1HE,1HT,1HU,1HR,1HN,1H /,             SIMP  6
     1 ((IALPH2(I),I=1,9)/1HC,1HO,1HN,1HT,1HI,1HN,1HU,1HE,1H /,        SIMP  7
     2 ((IALPH3(I),I=1,5)/1HS,1HT,1HO,1HP,1H /                         SIMP  8
     3 ,((IALPH4(I),I=1,10)/1HB,1HL,1HO,1HC,1HK,1HD,1HA,1HT,1HA,1H /   SIMP  9
      IF(ITYP .EQ. 10) GO TO 25                                         SIMP 10
      IF(ITYP .EQ. 7) GOTO 15                                           SIMP 11
      IF(ITYP .EQ. 29) GO TO 35                                         SIMP 12
      DO 10 I=1,7                                                       SIMP 13
      IF(NEXT(JPTR) .NE. IALPH1(I)) GO TO 50                            SIMP 14
   10 CONTINUE                                                          SIMP 15
      NB=1                                                              SIMP 16
      NBRNCH=1                                                          SIMP 17
      NBLOCK=NBLOCK+1                                                   SIMP 18
      IBLOCK(NBLOCK)=999                                                SIMP 19
      RETURN                                                            SIMP 20
   15 DO 20 I=1,9                                                       SIMP 21
      IF(NEXT(JPTR) .NE. IALPH2(I)) GO TO 50                            SIMP 22
   20 CONTINUE                                                          SIMP 23
      RETURN                                                            SIMP 24
   25 DO 30 I=1,5                                                       SIMP 25
      IF(NEXT(JPTR) .NE. IALPH3(I)) GO TO 50                            SIMP 26
   30 CONTINUE                                                          SIMP 27
      NB=1                                                              SIMP 28
      NBRNCH=1                                                          SIMP 29
      NBLOCK=NBLOCK+1                                                   SIMP 30
      IBLOCK(NBLOCK)=999                                                SIMP 31
      RETURN                                                            SIMP 32
   35 DO 40 I=1,10                                                      SIMP 33
      IF(NEXT(JPTR) .NE. IALPH4(I)) GO TO 50                            SIMP 34
   40 CONTINUE                                                          SIMP 35
      RETURN                                                            SIMP 36
   50 CALL ERROR(7,KDM1,KDM2)
      RETURN                                                            SIMP 38
      END                                                               SIMP 39
```

```
      SUBROUTINE SLEVEL(FAIL)
      COMMON/LVARGS/LVFUNC,LVVARG,LVVAD,LVVPOS,LVVTYP,LVVAL,
     +LVHEAD,LVVNVL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP
      COMMON/LVTABL/LVTSIZ,LVMAP(     1)/LVVSEQ/LVSIZE,LVSQSP(      1)
      COMMON /NEED/ START,ASSOC,LEVEL,STOP
      COMMON/NEEDS/STJ,JSTACK,R,JAS,J,JLAST,RTEMP,STACK(400,4)
      COMMON /STRING/ NNN(2),STR
      COMMON /JL/ JSTOP
      INTEGER START,STOP,ASSOC,STACK,STR(1),STJ,R,RTEMP
      LOGICAL FAIL
C     EXECUTE
      GO TO 25000
25001 CONTINUE
      RTEMP=0
      JSTOP=JSTACK
   10 IF(JSTOP .EQ. 0) GO TO 40
      NPNTR=STACK(JSTOP,4)
      IF(NPNTR .GT. 0) GO TO 20
      IF(STACK(JSTOP,2) .NE. 0) GO TO 30
      JSTOP=JSTOP-1
      GO TO 10
   20 JSTOP=NPNTR-1
      GO TO 10
   30 STACK(JSTACK,4)=JSTOP
      JAS=STACK(JSTOP,2)
      R=STACK(JSTOP,1)
      RTEMP=R
C     R+LEVEL.JAS @R
C****          R             +       LEVEL
      LVVPOS =         JAS
      LVVTYP =   3
      LVFUNC=       LEVEL
      LVVARG=          R
      CALL LVFIND(LV2       A,LV2       B,LV2       C,LV2       D)
      LV1    AAD =          R
      IF (LVVAL.NE.-1) LV1    AAD = LVVAL
C**** LV1    AAD           @            R
        R = LV1    AAD
      FAIL=.FALSE.
      IF(STACK(JSTOP,4) .LT. 0) STACK(JSTOP,4)=0
      RETURN
   40 FAIL=.TRUE.
      RETURN
C     COMPLETE
25000 CONTINUE
      LV2A=0
      LV2B=0
      LV2C=0
      LV2D=0
      GO TO 25001
      END
```

```
      SUBROUTINE SQUEEZ
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),   RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                   RICH   4
      INTEGER A,D,BLANK,AICH                                             SQUEEZE4
      DATA BLANK/1H /,AICH/1HH/                                          SQUEEZE5
      J=0                                                                SQUEEZE6
      DO 10 I=1,M                                                        SQUEEZE7
      IF(D(I) .EQ. BLANK) GO TO 10                                       SQUEEZE8
      J=J+1                                                              SQUEEZE9
      D(J)=D(I)                                                          SQUEEZ10
      IF(D(J) .NE. AICH) GO TO 10                                        SQUEEZ11
      IF(JTYP .NE. 3) GO TO 10                                           SQUEEZ12
      M=M+J-I                                                            SQUEEZ13
      RETURN                                                             SQUEEZ14
   10 CONTINUE                                                           SQUEEZ15
      M=J                                                                SQUEEZ16
      RETURN                                                             SQUEEZ17
      END                                                                SQUEEZ18
```

```
      SUBROUTINE SSTOP(FAIL)
      COMMON/LVARGS/LVFUNC,LVVARG,LVVAD,LVVPOS,LVVTYP,LVVAL,
     *LVHEAD,LVVNVL,LVDEST,LVVALS(10),LVTYPE(10),LVSKIP
      COMMON/LVTABL/LVTSIZ,LVMAP(     1)/LVVSEQ/LVSIZE,LVSQSP(     1)
      COMMON /NEED/ START,ASSOC,LEVEL,STOP
      COMMON/NEEDS/STJ,JSTACK,R,JAS,J,JLAST,RTEMP,STACK(400,4)
      COMMON /STRING/ NNN(2),STR
      INTEGER START,STOP,ASSOC,STACK,STR(1),STJ,R,TEMP
      LOGICAL FAIL
C     EXECUTE
      GO TO 25000
25001 CONTINUE
      JSTOPS=JSTACK
C    5 R=ASSOC/10
     5 CONTINUE
C**** R + ASSOC
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC= ASSOC
      LVVARG= R
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LVI AAD = R
      IF (LVVAL.NE.-1) LVI AAD = LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO 10
      JSTOPS=JSTOPS+1
      STACK(JSTOPS,1)=R
      STACK(JSTOPS,2)=0
      STACK(JSTOPS,4)=0
C    10 R(=STOP//20,+STOP/30 @R/5/5)
     10 CONTINUE
      LVI AAD = R
C**** LVI AAD = STOP
      LVVAL = -100
      IF (LVI AAD.NE. STOP) LVVAL = -1
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.NE.-1) GO TO 20
C**** LVI AAD + STOP
      LVVTYP = 3
      LVVPOS = 1
      LVINDX = 0
      LVFUNC= STOP
      LVVARG= LVI AAD
      CALL LVFIND(LVINDX,LVINDX,LVINDX,LVINDX)
      LVIAAI=LVIAAD
      IF(LVVAL .NE. -1) LVIAAI=LVVAL
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO 30
C**** LVI AAF @ R
      R=LVIAAI
      LVVTR = LVVAL
      LVVAL = -100
      IF (LVVTR.EQ.-1) GO TO 5
      IF (LVVTR.NE.-1) GO TO 5
   20 FAIL=.FALSE.
      RETURN
```

```
C  30 JSTACK=JSTOPS/40
   30 CONTINUE
C****     JSTACK            =       JSTOPS
          LVVAL = -100
          IF (    JSTACK.NE.    JSTOPS) LVVAL = -1
          LVVTR = LVVAL
          LVVAL = -100
          IF (LVVTR.EQ.-1) GO TO          40
          FAIL=.TRUE.
          RETURN
      40  R=STACK(JSTOPS,1)
          JAS=STACK(JSTOPS,2)+1
C         R+ASSOC,JAS @TEMP//50
C****           R               +       ASSOC
          LVVPOS =         JAS
          LVVTYP =    3
          LVFUNC=       ASSOC
          LVVARG=           R
          CALL LVFIND(LV2       A,LV2       B,LV2       C,LV2       D)
          LV1     AAD =          R
          IF (LVVAL.NE.-1) LV1     AAD = LVVAL
C**** LV1     AAD            @          TEMP
                TEMP = LV1     AAD
          LVVTR = LVVAL
          LVVAL = -100
          IF (LVVTR.NE.-1) GO TO          50
          JSTOPS=JSTOPS-1
C         //30
          LVVTR = LVVAL
          LVVAL = -100
          IF (LVVTR.NE.-1) GO TO          30
      50  STACK(JSTOPS,2)=JAS
C         R=TEMP//40
C****           R               =       TEMP
          LVVAL = -100
          IF (          R.NE.       TEMP) LVVAL = -1
          LVVTR = LVVAL
          LVVAL = -100
          IF (LVVTR.NE.-1) GO TO          40
C         TEMP @R//5
C****       TEMP            @          R
                R =        TEMP
          LVVTR = LVVAL
          LVVAL = -100
          IF (LVVTR.NE.-1) GO TO          5
C         COMPLETE
          RETURN
25000     CONTINUE
          LV2A=0
          LV2B=0
          LV2C=0
          LV2D=0
          GO TO 25001
          END
```

```
      SUBROUTINE STATNO                                                 STATNO 2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,NZ,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      COMMON/LABELS/STATRA(2,200),NLABEL                                STATNO 4
      COMMON/DOLOOP/ISTACK(4,50),NSTACK,ILOOP,IOVFL#
      COMMON/BASBLK/IBLOCK(2500),NBLOCK,NB,NBRNCH
      INTEGER A,BLANK,STATRA                                            STATNO 7
      INTEGER BITPUT,BITGET                                             STATNO 8
      DATA BLANK/1H /                                                   STATNO 9
      LOC=0                                                             STATN010
      DO 5 I=1,5                                                        STATN011
      IF(A(I) .NE. BLANK) GO TO 10                                      STATN012
    5 CONTINUE                                                          STATN013
      IF(ITYP .NE. 18) GO TO 7                                          STATN014
      IF(IBLKDT .EQ. 1) RETURN                                          STATN015
      IF(NBRNCH .EQ. 0) GO TO 110                                       STATN016
      IBLOCK(IBLKST)=BITPUT(IBLOCK(IBLKST),NBRNCH,6)                    STATN017
      RETURN                                                            STATN018
    7 IF(ITYP .EQ. 28) GO TO 90                                         STATN019
      IF(NBLOCK .EQ. 0) GO TO 8                                         STATN020
      IF(IBLOCK(NBLOCK) .EQ. 998) GO TO 32                              STATN021
      IF(NB .EQ. 2) GO TO 31                                            STATN022
      IF(NB .EQ. 1) GO TO 70                                            STATN023
      RETURN                                                            STATN024
    8 NBLOCK=1                                                          STATN025
      GO TO 34                                                          STATN026
   10 IF(ITYP .EQ. 18) GO TO 50                                         STATN027
      JPTR=1                                                            STATN028
      CALL GNLE                                                         STATN029
      IF(JTYP .NE. 5) GO TO 50                                          STATN030
      IF(A(6) .NE. BLANK) GO TO 50                                      STATN031
      IF(JPTR .LT. 6) GO TO 50                                          STATN032
      CALL STSRCH                                                       STATN033
      IF(BITGET(STATRA(2,LOC),9,3) .EQ. 1) GO TO 60                     STATN034
      STATRA(2,LOC)=BITPUT(STATRA(2,LOC),1,9)                           STATN035
      IF(LTYP .EQ. 9) GO TO 20                                          STATN036
      STATRA(2,LOC)=BITPUT(STATRA(2,LOC),ITYP,6)                        STATN037
      GO TO 30                                                          STATN038
   20 STATRA(2,LOC)=BITPUT(STATRA(2,LOC),9,6)                           STATN039
   30 IF(ITYP .EQ. 28) RETURN                                           STATN040
      IF(NBLOCK .EQ. 0) GO TO 8                                         STATN041
      IF(NB .EQ. 1) GO TO 32                                            STATN042
   31 NBLOCK=NBLOCK+1                                                   STATN043
      IBLOCK(NBLOCK)=998                                                STATN044
      NBRNCH=1                                                          STATN045
   32 NB=0                                                              STATN046
      NBLOCK=NBLOCK+1                                                   STATN047
      IBLOCK(IBLKST)=BITPUT(IBLOCK(IBLKST),NBLOCK,28)                   STATN048
      IBLOCK(IBLKST)=BITPUT(IBLOCK(IBLKST),NBRNCH,6)                    STATN049
      J1=IBLKST+1                                                       STATN050
      J2=NBLOCK-NBRNCH-1                                                STATN051
      IF(IBLOCK(J2) .GT. 6000) GO TO 34                                 STATN052
      J21=J2-1                                                          STATN053
   21 IF(IBLOCK(J1) .LT. 6000) GO TO 34                                 STATN054
      IRES=IBLOCK(J1)                                                   STATN055
      DO 22 K2=J1,J21                                                   STATN056
   22 IBLOCK(K2)=IBLOCK(K2+1)                                           STATN057
      IBLOCK(J2)=IRES                                                   STATN058
      GO TO 21                                                          STATN059
```

```
 34 IBLKST=NBLOCK                                                        STATN060
    NBRNCH=0                                                             STATN061
    IBLOCK(IBLKST)=BITPUT(LOC,ILOOP,12)                                  STATN062
    IF(LOC .EQ. 0) RETURN                                                STATN063
    STATRA(2,LOC)=BITPUT(STATRA(2,LOC),IBLKST,36)                        STATN064
    IF(BITGET(STATRA(2,LOC),15,3) .NE. 1) RETURN                         STATN065
    IF(IOVFL# .EQ. 1) RETURN
    IF(LOC .NE. ISTACK(1,ILOOP)) GO TO 80                                STATN066
    IF(ITYP .GE. 3 .AND. ITYP .LE. 6) GO TO 100                          STATN067
    IF( ITYP .EQ. 9 .OR. ITYP .EQ. 10 .OR. ITYP .EQ. 17) GO TO 100       STATN068
    NB=2                                                                 STATN069
    ISTACK(2,ILOOP)=1                                                    STATN070
    NBLOCK=NBLOCK+1                                                      STATN071
    IBLOCK(NBLOCK)=6000+ISTACK(4,ILOOP)                                  STATN072
    KLOOP=ILOOP-1                                                        STATN073
    DO 40 J=1,KLOOP                                                      STATN074
    LOOP=ILOOP-J                                                         STATN075
    IF(ISTACK(2,LOOP) .EQ. 1) GO TO 40                                   STATN076
    IF(ISTACK(1,LOOP) .EQ. LOC) GO TO 35                                 STATN077
    ILOOP=LOOP                                                           STATN078
    RETURN                                                               STATN079
 35 ISTACK(2,LOOP)=1                                                     STATN080
    NBLOCK=NBLOCK+1                                                      STATN081
    IBLOCK(NBLOCK)=6000+ISTACK(4,LOOP)                                   STATN082
 40 CONTINUE                                                             STATN083
    ILOOP=0                                                              STATN084
    RETURN                                                               STATN085
 50 IERC=32
    GO TO 200
 60 IERC=33
    GO TO 200
 70 IERC=34
    GO TO 200
 80 IERC=35
    GO TO 200
 90 IERC=36
    GO TO 200
100 IERC=37
    GO TO 200
110 IERC=38
200 CALL ERROR(IERC,KDM1,KDM2)
    RETURN                                                               STATN100
    END                                                                  STATN101
```

```
      SUBROUTINE STFNC(NFNC)
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/FUNC/IFNCRA(5,17),MARGS,IARGS(50),FNCLOC(5),NFUNC
      INTEGER BITGET
      NARG=BITGET(IDTBL(3,LOC),7,6)
      NAR2=IFNCRA(NFNC,1)
      IF(NARG .NE. NAR2) CALL ERROR(26,IDTBL(1,LOC),KDM2)
      NARGS=MIND(NARG,NAR2)
      KOUNT=0
      NT=1+(NARG-1)/4
      DO 10 I=1,NT
      ICOL1=-6
      ICOL2=-3
      DO 10 J=1,4
      KOUNT=KOUNT+1
      IF(KOUNT .GT. NARGS) RETURN
      ICOL1=ICOL1+9
      ICOL2=ICOL2+9
      IF(BITGET(IFNCRA(NFNC,I+1),ICOL2,3) .NE. 0) CALL ERROR(50,KOUNT,K)
      ITP1=BITGET(IFNCRA(NFNC,I+1),ICOL1,3)
      IF(ITP1 .EQ. 0) GO TO 10
      ITP2=BITGET(IDTBL(3,LOC+KOUNT),10,3)
      IF(ITP1 .NE. ITP2) CALL ERROR(51,KOUNT,KDM2)
   10 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE STORE                                                  STORE  2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      NID=NID+1                                                         STORE  4
      IF(NID .GT. 500) GO TO 20
      IF(INITID(IDTYP) .NE. 0) GO TO 5                                  STORE  5
      INITID(IDTYP)=NID                                                 STORE  6
    5 CONTINUE                                                          STORE  7
      IDTBL(1,NID)=NXTID                                                STORE  8
      IDTBL(2,NID)=0                                                    STORE  9
      IF(LASTID(IDTYP) .EQ. 0) GO TO 10                                 STORE 10
      LAST=LASTID(IDTYP)
      IDTBL(2,LAST)=NID
   10 LASTID(IDTYP)=NID                                                 STORE 12
      RETURN                                                            STORE 13
   20 WRITE(6,25)
   25 FORMAT(/////5X,46H SYMBOL TABLE OVERFLOW - PROCESSING TERMINATED)
      STOP
      END                                                               STORE 14
```

```
      SUBROUTINE STSRCH                                                 STSRCH 2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
     • JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH   4
      COMMON/LABELS/STATRA(2,200),NLABEL                                STSRCH 4
      INTEGER STATRA                                                    STSRCH 5
      IF(NLABEL .EQ. 0) GO TO 15                                        STSRCH 6
      DO 10 I=1,NLABEL                                                  STSRCH 7
      IF(STATRA(1,I) .NE. N2) GO TO 10                                  STSRCH 8
      LOC=I                                                             STSRCH 9
      RETURN                                                            STSRCH10
   10 CONTINUE                                                          STSRCH11
   15 NLABEL=NLABEL+1                                                   STSRCH12
      IF(NLABEL .GT. 200) GO TO 20
      LOC=NLABEL                                                        STSRCH13
      STATRA(1,LOC)=N2                                                  STSRCH14
      RETURN                                                            STSRCH15
   20 WRITE(6,25)
   25 FORMAT(/////5X,53H STATEMENT NO. TABLE OVERFLOW - PROCESSING TERMI
     •NATED)
      STOP
      END                                                               STSRCH16
```

```
      SUBROUTINE SUB                                                    SUB   2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH  2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH  4
      DIMENSION IALPH(10),IALPH2(8),KT(5)                               SUB   4
      COMMON /SUBS/ISUBS(3,100),NSUBS                                   SUB   5
      INTEGER BLANK,COMMA,RPAR,A,D                                      SUB   6
      INTEGER BITPUT                                                    SUB   7
      DATA (IALPH(I),I=1,10)/1HS,1HU,1HB,1HR,1HO,1HU,1HT,1HI,1HN,1HE/   SUB   8
      DATA (IALPH2(I),I=1,8)/1HF,1HU,1HN,1HC,1HT,1HI,1HO,1HN/           SUB   9
      DATA (KT(I),I=1,5)/1HR,1HC,1HO,1HI,1HL/                           SUB  10
      DATA BLANK/1H /,LPAR/1H(/,RPAR/1H)/,COMMA/1H,/                    SUB  11
      NARG=0                                                            SUB  12
      ISTATE=0
      IT=32-ITYP                                                        SUB  15
      GO TO (2,5),IT                                                    SUB  16
    2 DO 3 I=1,8                                                        SUB  17
      IF(NEXT(JPTR) .NE. IALPH2(I)) GO TO 50                            SUB  18
    3 CONTINUE                                                          SUB  19
      GO TO 12                                                          SUB  21
    5 DO 10 I=1,10                                                      SUB  22
      IF(NEXT(JPTR) .NE. IALPH(I)) GO TO 50                             SUB  23
   10 CONTINUE                                                          SUB  24
      GO TO 17                                                          SUB  26
   12 IPTR=JPTR                                                         SUB  27
      IFRST=NEXT(1)                                                     SUB  28
      IF(IFRST .EQ. 1HF) GO TO 14                                       SUB  29
      DO 13 I=1,5                                                       SUB  30
      IF(IFRST .NE. KT(I)) GO TO 13                                     SUB  31
      ISTATE=I
      GO TO 14                                                          SUB  33
   13 CONTINUE                                                          SUB  34
   14 JPTR=IPTR                                                         SUB  35
   17 CALL GNLE                                                         SUB  36
      IF(JTYP .NE. 2) GO TO 50                                          SUB  37
      IDTYP=2                                                           SUB  39
      CALL STORE                                                        SUB  40
      IE=D(1)
      IF(ITYP .NE. 31) GO TO 15                                         SUB  42
      IF(NEXT(JPTR) .NE. LPAR) GO TO 50                                 SUB  43
      IFNCNM=NXTID                                                      SUB  44
      GO TO 20                                                          SUB  45
   15 IF(NEXT(JPTR) .EQ. BLANK) GO TO 30                                SUB  46
      IF(A(JPTR-1) .NE. LPAR) GO TO 50                                  SUB  47
   20 CALL GNLE                                                         SUB  48
      IF(JTYP .NE. 2) GO TO 60                                          SUB  49
      CALL SEARCH                                                       SUB  50
      IF(ISRCH(1) .NE. 0 .OR. ISRCH(2) .NE. 0) CALL ERROR(86,NXTID,KDM2)
      IDTYP=1                                                           SUB  52
      CALL STORE                                                        SUB  53
      IDTBL(3,NID)=BITPUT(IDTBL(3,NID),1,12)                            SUB  54
      NARG=NARG+1                                                       SUB  55
      IF(NEXT(JPTR) .EQ. RPAR) GO TO 30                                 SUB  56
      IF(A(JPTR-1) .NE. COMMA) GO TO 50                                 SUB  57
      GO TO 20                                                          SUB  58
   30 LOC=1
      D(1)=IE
      IF(NARG .GT. 63) CALL ERROR(83,KDM1,KDM2)
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),NARG,7)                          SUB  61
      IF(ITYP .EQ. 30) RETURN
      IF(ISTATE .EQ. 0) GO TO 55
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),ISTATE,10)                       SUB  66
      IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,11)                            SUB  67
      RETURN                                                            SUB  68
   55 CALL IMPTYP
      RETURN
   50 CALL ERROR(7,KDM1,KDM2)
      RETURN                                                            SUB  70
   60 CALL ERROR(86,NXTID,KDM2)
      RETURN                                                            SUB  72
      END                                                               SUB  73
```

```
      SUBROUTINE SUBCHK
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES
      COMMON/GLOBAL/NBLK,NREF,NSUBS,BLKTBL(200),EXTTBL(100),ISUBS(100)
      COMMON/LIST/NLIST,NINTFC,ISUBLT(3,200),INTFAC(500)
      INTEGER BITPUT,BITGET
      IF(IBLKDT .EQ. 1) RETURN
      NSUBS=NSUBS+1
      IF(NSUBS .GT. 100) GO TO 50
      NARG=BITGET(IDTBL(3,I),7,6)
      ITP=BITGET(IDTBL(3,I),10,3)
      IF(BITGET(IDTBL(3,I),18,1) .EQ. 1) GO TO 15
      IDTBL(3,I)=BITPUT(IDTBL(3,I),1,18)
      DO 5 I=1,NLIST
      IF(IDTBL(1,1) .NE. ISUBLT(1,I)) GO TO 5
      LISTLC=I
      IDTBL(3,I)=BITPUT(IDTBL(3,I),LISTLC,36)
      GO TO 20
    5 CONTINUE
      CALL ERROR(52,KDM1,KDM2)
      NLIST=NLIST+1
      ISUBLT(1,NLIST)=IDTBL(1,I)
      ISUBS(NSUBS)=NLIST
      ISUBLT(2,NLIST)=0
      ISUBLT(3,NLIST)=0
      IDTBL(3,I)=BITPUT(IDTBL(3,I),NLIST,36)
      IF(NARG .EQ. 0) RETURN
      IPTR=NINTFC+1
      ISUBLT(2,NLIST)=BITPUT(0,NARG,6)
      ISUBLT(2,NLIST)=BITPUT(ISUBLT(2,NLIST),ITP,13)
      ISUBLT(3,NLIST)=IPTR
      NINTFC=IPTR+(NARG-1)/4
      KOUNT=0
      DO 10 I=IPTR,NINTFC
      INTFAC(I)=0
      ICOL1=-6
      ICOL2=-3
      DO 10 J=1,4
      KOUNT=KOUNT+1
      IF(KOUNT .GT. NARG) RETURN
      ICOL1=ICOL1+9
      ICOL2=ICOL2+9
      ITP=BITGET(IDTBL(3,KOUNT+1),10,3)
      NDIM=BITGET(IDTBL(3,KOUNT+1),7,6)
      IDTBL(3,KOUNT+1)=BITPUT(IDTBL(3,KOUNT+1),1,15)
      INTFAC(I)=BITPUT(INTFAC(I),ITP,ICOL1)
   10 INTFAC(I)=BITPUT(INTFAC(I),NDIM,ICOL2)
      RETURN
   15 LISTLC=BITGET(IDTBL(3,I),36,9)
   20 ISUBS(NSUBS)=LISTLC
      KLAS=BITGET(ISUBLT(2,LISTLC),10,4)
      NAR2=BITGET(ISUBLT(2,LISTLC),6,6)
      IF(NARG .NE. NAR2) CALL ERROR(26,IDTBL(1,1),KDM2)
      NARGS=MIN0(NARG,NAR2)
      IF(ITP .NE. BITGET(ISUBLT(2,LISTLC),13,3))
     $ CALL ERROR(49,IDTBL(1,1),KDM2)
      IF(NARGS .EQ. 0) RETURN
      IPTR=ISUBLT(3,LISTLC)
      NDPTR=IPTR+(NARGS-1)/4
```

```
61•           KOUNT=0
62•           DO 25 I=IPTR,NDPTR
63•           ICOL1=-6
64•           ICOL2=-3
65•           DO 25 J=1,4
66•           KOUNT=KOUNT+1
67•           IF(KOUNT .GT. NARGS) RETURN
68•           ICOL1=ICOL1+9
69•           ICOL2=ICOL2+9
70•           ITP=BITGET(INTFAC(I),ICOL1,3)
71•           NDIM=BITGET(INTFAC(I),ICOL2,3)
72•           ITP2=BITGET(IDTBL(3,KOUNT+1),10,3)
73•           NDIM2=BITGET(IDTBL(3,KOUNT+1),7,6)
74•           IOSTAT=BITGET(INTFAC(I),ICOL2+2,2)
75•           IF(IOSTAT .EQ. 2 .OR. KLAS .EQ. 0) IOSTAT=1
76•           IDTBL(3,KOUNT+1)=BITPUT(IDTBL(3,KOUNT+1),IOSTAT,15)
77•           IF(NDIM .NE. NDIM2) CALL ERROR(50,KOUNT,KDM2)
78•           IF(ITP2 .NE. 0) GO TO 23
79•           ITP2=1
80•           IFST=BITGET(IDTBL(1,KOUNT+1),6,6)
81•           IF(IFST .LE. 19 .AND. IFST .GE. 14) ITP2=4
82•        23 IF(ITP .NE. ITP2) CALL ERROR(51,KOUNT,KDM2)
83•        25 CONTINUE
84•           RETURN
85•        50 CALL ERROR(25,KDM1,KDM2)
86•           STOP
87•           END
```

```
1•            SUBROUTINE SWITCH                                          SWITCH  2
2•            COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH   2
3•           • JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
4•           2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                           RICH   4
5•            DO 20 I=1,NID                                              SWITCH  4
6•            IF(IDTBL(2,I) .NE. LOC) GO TO 20                           SWITCH  5
7•            IDTBL(2,I)=IDTBL(2,LOC)                                    SWITCH  6
8•            IF(LASTID(1) .EQ. LOC) LASTID(1)=I                         SWITCH  7
9•            GO TO 30                                                   SWITCH  8
10•        20 CONTINUE                                                   SWITCH  9
11•           INITID(1)=IDTBL(2,LOC)                                     SWITCH10
12•        30 LAST=LASTID(2)
13•           IDTBL(2,LAST)=LOC
14•           IDTBL(2,LOC)=0                                             SWITCH12
15•           LASTID(2)=LOC                                              SWITCH13
16•           CALL ERROR(87,IDTBL(1,LOC),KDM2)
17•           RETURN                                                     SWITCH14
18•           END                                                        SWITCH15
```

```
      SUBROUTINE SYMTAB                                                 SYMTAB  2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),  RICH    2
     • JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                  RICH    4
      COMMON/LABELS/STATRA(2,200),NLABEL                                SYMTAB  4
      COMMON/LIST/NLIST,NINTFC,ISUBLT(3,200),INTFAC(500)
      COMMON/STFUNC/NSTFNC,ISTFNC(10)
      DIMENSION ITABEL(5)                                               SYMTAB  5
      INTEGER TYPE(5),DIMS(3),BITGET,STATRA                             SYMTAB  6
      DATA (TYPE(I),I=1,5)/4HREAL,6HCOMPLX,6HDOUBLE,6HINTEGR,6HLOGICL/  SYMTAB  7
      DATA (DIMS(I),I=1,3)/1H1,1H2,1H3/                                 SYMTAB  8
      IF(NID .LE. 1) RETURN                                             SYMTAB  9
      IEXT=0                                                            SYMTAB10
      IF(IBLKDT .EQ. 1) GO TO 2                                         SYMTAB11
      INTL=INITID(2)
      WRITE(6,1) IDTBL(1,INTL)
    1 FORMAT(//////45X,25H SYMBOL TABLE FOR MODULE ,A6)                 SYMTAB13
      GO TO 4                                                           SYMTAB14
    2 WRITE(6,3)                                                        SYMTAB15
    3 FORMAT(//////46X,28H SYMBOL TABLE FOR BLOCK DATA)                 SYMTAB16
    4 LOC=INITID(1)                                                     SYMTAB17
      IF(LOC .EQ. 0) GO TO 28                                           SYMTAB18
      WRITE(6,5)                                                        SYMTAB19
    5 FORMAT(//56X,9HVARIABLES/30X,4HNAME,12X,4HTYPE,31X,10HRELOCATION) SYMTAB20
  100 ITABEL(1)=IDTBL(1,LOC)                                            SYMTAB21
      IF(BITGET(IDTBL(3,LOC),11,1) .EQ. 0) GO TO 27
      I=BITGET(IDTBL(3,LOC),10,3)
      ITABEL(2)=TYPE(I)                                                 SYMTAB23
   12 IF(BITGET(IDTBL(3,LOC),1,1) .EQ. 0) GO TO 16
      ITABEL(3)=5HARRAY                                                 SYMTAB26
      I=BITGET(IDTBL(3,LOC),7,6)                                        SYMTAB27
      ITABEL(4)=DIMS(I)                                                 SYMTAB28
      GO TO 18                                                          SYMTAB29
   16 ITABEL(3)=1H                                                      SYMTAB30
      ITABEL(4)=1H                                                      SYMTAB31
   18 IF(BITGET(IDTBL(3,LOC),12,1) .EQ. 0) GO TO 20                     SYMTAB32
      ITABEL(5)=5HF. P.                                                 SYMTAB33
      GO TO 25                                                          SYMTAB34
   20 IF(BITGET(IDTBL(3,LOC),16,1) .EQ. 1) GO TO 22                     SYMTAB35
      ITABEL(5)=1H                                                      SYMTAB36
      GO TO 25                                                          SYMTAB37
   22 ICOMNM=IDTBL(6,LOC)                                               SYMTAB38
      ITABEL(5)=IDTBL(1,ICOMNM)                                         SYMTAB39
      IF(ITABEL(5) .EQ. 1H ) ITABEL(5)=2H//                             SYMTAB40
   25 WRITE(6,26) (ITABEL(I),I=1,5)                                     SYMTAB41
   26 FORMAT(30X,A6,11X,A6,14X,A5,1X,A1,7X,A6)                          SYMTAB42
   27 LOC=IDTBL(2,LOC)
      IF(LOC .NE. 0) GO TO 100                                          SYMTAB44
   28 IF(IBLKDT .EQ. 1) GO TO 60
      LOC=IDTBL(2,INTL)
      IF(LOC .EQ. 0) GO TO 60
      WRITE(6,31)                                                       SYMTAB48
   31 FORMAT(//55X,10H EXTERNALS/44X,4HNAME,10X,4HTYPE,10X,4HARGS)      SYMTAB49
   30 ITABEL(1)=IDTBL(1,LOC)
      LISTLC=BITGET(IDTBL(3,LOC),36,9)
      IF(LISTLC .EQ. 0) GO TO 39
      ITP=BITGET(ISUBLT(2,LISTLC),13,3)
      IF(ITP .EQ. 0) GO TO 32
      ITABEL(2)=TYPE(ITP)
      GO TO 35
```

```
      32 ITABEL(2)=1H
      35 IF(BITGET(ISUBLT(2,LISTLC),14,1) .EQ. 1) GO TO 37
         ITABEL(3)=BITGET(ISUBLT(2,LISTLC),6,6)
         WRITE(6,36) (ITABEL(I),I=1,3)                                  SYMTAB55
      36 FORMAT(99X,A6,8X,A6,8X,I2)                                     SYMTAB56
         GO TO 39
      37 WRITE(6,38) ITABEL(1),ITABEL(2)
      38 FORMAT(99X,A6,8X,A6,8X,2H I)
      39 LOC=IDTBL(2,LOC)
         IF(LOC .NE. 0) GO TO 30                                        SYMTAB58
      40 IF(NSTFNC .EQ. 0) GO TO 90
         WRITE(6,62)
      62 FORMAT(//50X,20H STATEMENT FUNCTIONS/
        $ 99X,9HNAME,10X,9HTYPE,10X,9HARGS)
         DO 70 I=1,NSTFNC
         LC=ISTFNC(I)
         ITP=BITGET(IDTBL(3,LC),10,3)
         NRG=BITGET(IDTBL(3,LC),7,6)
      70 WRITE(6,36) IDTBL(1,LC),TYPE(ITP),NRG
      80 IF(NLABEL .EQ. 0) GO TO 50                                     SYMTAB59
         WRITE(6,92)                                                    SYMTAB60
      82 FORMAT(//51X,17H STATEMENT LABELS)                             SYMTAB61
         WRITE(6,95) (STATRA(1,I),I=1,NLABEL)                           SYMTAB62
      84 FORMAT(40X,518)                                                SYMTAB63
      85 DO 97 I=1,NLABEL                                               SYMTAB64
         IF(BITGET(STATRA(2,I),9,3) .NE. 1) CALL ERROR(15,STATRA(1,I),KDM2)
         IF(BITGET(STATRA(2,I),12,3) .NE. 1)CALL ERROR(16,STATRA(1,I),KDM2)
      87 CONTINUE                                                       SYMTAB67
      89 LOC=INITIO(3)                                                  SYMTAB70
         IF(LOC .EQ. 0) RETURN                                          SYMTAB71
         WRITE(6,52)                                                    SYMTAB72
      92 FORMAT(//53X,14H COMMON BLOCKS/50X,9HNAME,10X,6HLENGTH)        SYMTAB73
      93 ITABEL(1)=IDTBL(1,LOC)                                         SYMTAB74
         IF(ITABEL(1) .EQ. 1H ) ITABEL(1)=2H//                          SYMTAB75
         WRITE(6,55) ITABEL(1),IDTBL(4,LOC)                             SYMTAB76
      96 FORMAT(50X,A6,8X,I8)                                           SYMTAB77
         LOC=IDTBL(2,LOC)                                               SYMTAB78
         IF(LOC .NE. 0) GO TO 51                                        SYMTAB79
         RETURN                                                         SYMTAB80
         END                                                            SYMTAB81
```

```
      SUBROUTINE TYPE                                                    TYPE   2
      COMMON A(1326),D(500),IDTBL(8,500),INITID(3),LASTID(3),ISRCH(3),   RICH   2
     * JPTR,N,M,JTYP,LSTART,N2,IFNCNM,LOGID,NXTID,IDTYP,NID,LOC,
     2 LTYP,ITYP,IBLKDT,MODE,IERR,IDES                                   RICH   4
      DIMENSION IALPH1(7),IALPH2(7),IALPH3(9),IALPH4(7),IALPH5(15),      TYPE   4
     1 IDIM(3)                                                           TYPE   5
      INTEGER A,RPAR,COMMA,BLANK                                         TYPE   6
      INTEGER BITPUT,BITGET,COMLOC                                       TYPE   7
      DATA (IALPH1(I),I=1,7)/1HL,1HO,1HG,1H1,1HC,1HA,1HL/,               TYPE   8
     1 (IALPH2(I),I=1,7) /1HI,1HN,1HT,1HE,1HG,1HE,1HR/,                 TYPE   9
     2 (IALPH3(I),I=1,9)/1HR,1HE,1HA,1HL/,                              TYPE  10
     3 (IALPH4(I),I=1,7)/1HC,1HO,1HM,1HP,1HL,1HE,1HX/,                  TYPE  11
     4 (IALPH5(I),I=1,15)/1HO,1HO,1HU,1HB,1HL,1HE,1HP,1HR,1HE,1HC,      TYPE  12
     5 1HI,1HS,1HI,1HO,1HN/                                             TYPE  13
      DATA LPAR/1H(/,RPAR/1H)/,COMMA/1H,/,BLANK/1H /                     TYPE  14
      MUL=1                                                              TYPE  15
      IT=ITYP-18                                                         TYPE  16
      GO TO (10,20,30,90,50),IT                                          TYPE  17
   10 DO 15 I=1,7                                                        TYPE  18
      IF(NEXT(JPTR) .NE. IALPH2(I)) GO TO 110                            TYPE  19
   15 CONTINUE                                                           TYPE  20
      ISTATE=3                                                           TYPE  21
      ISTATE=9
      GO TO 60                                                           TYPE  22
   20 DO 25 I=1,9                                                        TYPE  23
      IF(NEXT(JPTR) .NE. IALPH3(I)) GO TO 110                            TYPE  24
   25 CONTINUE                                                           TYPE  25
      ISTATE=1
      GO TO 60                                                           TYPE  27
   30 DO 35 I=1,15                                                       TYPE  28
      IF(NEXT(JPTR) .NE. IALPH5(I)) GO TO 110                            TYPE  29
   35 CONTINUE                                                           TYPE  30
      MUL=2                                                              TYPE  31
      ISTATE=3
      GO TO 60                                                           TYPE  33
   90 DO 95 I=1,7                                                        TYPE  34
      IF(NEXT(JPTR) .NE. IALPH4(I)) GO TO 110                            TYPE  35
   95 CONTINUE                                                           TYPE  36
      MUL=2                                                              TYPE  37
      ISTATE=2
      GO TO 60                                                           TYPE  39
   50 DO 55 I=1,7                                                        TYPE  40
      IF(NEXT(JPTR) .NE. IALPH1(I)) GO TO 110                            TYPE  41
   55 CONTINUE                                                           TYPE  42
      ISTATE=5
   60 ISUB=0                                                             TYPE  44
      INCR=MUL                                                           TYPE  45
      CALL GNLE                                                          TYPE  46
      IF(JTYP .NE. 2) GO TO 110                                          TYPE  47
      CALL SEARCH                                                        TYPE  48
      IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID,KDM2)
      IF(ISRCH(1) .EQ. 1) GO TO 62                                       TYPE  50
      IDTYP=1                                                            TYPE  51
      CALL STORE                                                         TYPE  52
      LOC=NID                                                            TYPE  53
   62 IF(BITGET(IDTBL(3,LOC),11,1) .NE. 0) GO TO 120
      IF(NEXT(JPTR) .NE. LPAR) GO TO 87                                  TYPE  55
      ISUB=1                                                             TYPE  56
      IE=LOC                                                             TYPE  57
      I=0
```

```
     68 I=I+1
        CALL GNLE                                                    TYPE  59
        IF(JTYP .NE. 5) GO TO 65                                     TYPE  60
        IDIM(I)=N2                                                   TYPE  61
        IF(N2 .GT. (2**17-1)) CALL ERROR(8,KDM1,KDM2)
        IF(N2 .LE. 0) CALL ERROR(8,KDM1,KDM2)
        INCR=INCR*N2                                                 TYPE  64
        GO TO 67                                                     TYPE  65
     65 IF(JTYP .NE. 2) GO TO 110                                    TYPE  66
        CALL SEARCH                                                  TYPE  67
        IF(ISRCH(2) .EQ. 1) CALL ERROR(10,NXTID,KDM2)
        IF(ISRCH(1) .EQ. 1) GO TO 66                                 TYPE  69
        IDTYP=1                                                      TYPE  70
        CALL STORE                                                   TYPE  71
        LOC=NID                                                      TYPE  72
     66 IF(BITGET(IDTBL(3,LOC),12,1) .NE. 1) CALL ERROR(9,KDM1,KDM2)
        IF(BITGET(IDTBL(3,LOC),1,1) .NE. 0) GO TO 130
        CALL IMPTYP                                                  TYPE  75
        IF(BITGET(IDTBL(3,LOC),10,3) .NE. 4) CALL ERROR(9,KDM1,KDM2)
        IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,13)                       TYPE  77
        IDIM(I)=2**17+LOC                                            TYPE  78
     67 IF(NEXT(JPTR) .EQ. COMMA) GO TO 68                           TYPE  79
        IF(A(JPTR-1) .NE. RPAR) GO TO 110                            TYPE  80
        K=NEXT(JPTR)                                                 TYPE  81
        LOC=IE
        IF(BITGET(IDTBL(3,LOC),1,1) .NE. 0)
       $ CALL ERROR(11,IDTBL(1,LOC),KDM2)
        IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,1)                        TYPE  86
        IF(I .GT. 3) GO TO 110                                       TYPE  87
        GO TO (85,80,75),I                                           TYPE  88
     75 IDTBL(4,LOC)=BITPUT(IDTBL(4,LOC),IDIM(3),36)                 TYPE  89
     80 IDTBL(4,LOC)=BITPUT(IDTBL(4,LOC),IDIM(2),18)                 TYPE  90
     85 IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),IDIM(1),36)                 TYPE  91
        IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),I,7)                        TYPE  92
     87 IF(INCR .EQ. 1) GO TO 90                                     TYPE  93
        IF(BITGET(IDTBL(3,LOC),16,1) .NE. 1) GO TO 90                TYPE  94
        COMLOC=IDTBL(6,LOC)                                          TYPE  95
        IDTBL(4,COMLOC)=IDTBL(4,COMLOC)+INCR-1                       TYPE  96
     90 IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),ISTATE,10)                  TYPE  97
        IDTBL(3,LOC)=BITPUT(IDTBL(3,LOC),1,11)                       TYPE  98
        IF(A(JPTR-1) .EQ. COMMA) GO TO 60                            TYPE  99
        IF(NEXT(JPTR) .EQ. BLANK) RETURN                             TYPE 100
    110 CALL ERROR(7,KDM1,KDM2)
        RETURN                                                       TYPE 102
    120 CALL ERROR(12,NXTID,KDM2)
        RETURN
    130 CALL ERROR(14,NXTID,KDM2)
        RETURN
        END                                                          TYPE 103
```

Auxiliary Programs and Associated Data

Program GRAPH

```
      DIMENSION INT(1000)
   10 FORMAT(12I6)
   20 FORMAT(2I6,A6,I6,A6,7I6/A6,2I6,A6,3I6,A6,I6,A6,2I6/A6,4I6,A6,3I6,
     $ A6,2I6/A6,2I6,A6,8I6/A6,I6,A6,I6,A6,I6,A6,3I6,2A6/3I6,A6,3I6,A6,
     $ 3I6,A6/7I6,A6,4I6/5I6,A6,6I6/(12I6))
      READ 10,(INT(I),I=1,11)
      WRITE(4) (INT(I),I=1,11)
      READ 20,INT
      WRITE(4) INT
      READ 10,INT
      WRITE(4) INT
      READ 10,INT
      WRITE(4) INT
      READ 10,INT
      WRITE(4) INT
      READ 10,INT(1)
      WRITE(4) INT(1)
      READ 10,(INT(I),I=1,34)
      WRITE(4) (INT(I),I=1,34)
      STOP
      END
```

Syntax Graph

```
  1000    99    17     8    17     0    21    74     0     0     1
   359    60+          1-           2   103   103    67     3     3     4
•         72    5RL          0   669   483,          10CP       10000   349
•          7   480   795   795DP      20000    21     8)           9    42
          18   480'        930    10   935     6   103   935   480   256
C         120         235          13'          14   346   47C     OR
    15   795   795>          16   391   411NT        17   322   359X
    19    28    47   359    29   935   977>        110    67   318   318
    29   228   359    40   935/         130     3    72    11    72    60
    29    60   999    99   100   101   102   103   104   105   106    40
   107    62   109   111   112   113   114   115   116   117   118   119
    10   120   122   123   124   125   126   127    81    62   128    21
   131   133   134   135   136   137   157   138    93   140   142   143
   144   145   146   147    81   480   148   151   152   153   154   155
   156   110   157   159    93   114   160   163   164   165   166   167
   168   285    60   169   172   173    86   174    88   130   176   179
   133   114    72   180   184   185   186   187   188   189   516   190
   192   193   194   195    86   130   196   199   133   200   202   203
   204   205   206   207   208   209   210   211   212   103   126   208
```

213 217 208 218 220 221 222 223 224 710 225 553
227 182 229 231 480 292 232 235 126 236 150 238
566 240 242 243 244 157 245 247 248 182 249 251
252 253 254 255 256 257 258 318 150 259 262 175
263 691 265 593 267 269 270 271 272 273 274 228
275 277 278 279 280 281 282 283 284 175 285 287
288 289 290 291 292 293 294 228 295 297 298 299
300 301 302 256 630 303 306 307 378 308 263 310
312 313 314 315 316 317 208 480 318 321 322 346
323 325 326 327 328 329 263 330 332 333 334 288
335 337 338 339 340 341 342 343 670 235 344 347
348 349 350 351 352 353 354 288 355 357 358 418
359 361 315 362 364 365 366 367 368 322 369 371
372 285 373 375 376 377 378 379 380 381 315 382
384 711 385 387 388 322 389 391 392 3 393 285
395 397 398 399 353 400 292 402 404 405 406 407
408 409 410 411 412 346 413 415 416 417 418 419
353 420 422 423 424 425 426 753 318 427 430 431
385 432 86 434 436 437 438 439 440 441 442 443
444 445 446 447 896 448 450 451 385 452 454 455
346 456 110 458 460 461 462 463 464 465 378 466
468 359 796 469 472 473 474 475 476 477 411 478
480 133 491 493 484 485 486 487 378 488 490 491
445 492 494 495 555 496 498 499 500 501 502 503
504 505 506 507 508 509 510 511 445 512 840 514
516 517 518 519 520 521 522 523 524 525 519 526
418 528 483 530 532 533 487 534 536 537 538 539
452 540 542 28 543 545 546 547 548 549 483 550
552 553 497 445 554 557 558 885 559 561 452 562
564 565 566 567 568 569 570 571 572 573 574 60
529 575 578 579 580 581 582 583 584 585 586 72
587 589 480 590 592 483 593 595 529 596 208 598
600 601 602 88 603 931 605 519 607 609 610 611
612 613 614 615 942 616 618 619 620 621 622 623
624 625 626 627 628 629 630 631 632 633 634 635
636 527 637 639 640 641 642 643 644 645 646 647
648 649 650 651 978 652 654 655 982 656 658 659
660 661 662 663 664 665 666 667 668 669 322 670
672 673 674 675 676 677 678 679 680 681 8 682
684 11 685 687 688 689 690 603 691 693 694 695
696 697 698 699 700 701 702 703 704 705 706 707
708 157 709 711 712 603 713 715 669 716 718 719
720 721 722 723 724 725 726 727 728 729 730 731
732 733 734 735 669 736 738 739 740 741 742 743
744 745 746 747 748 749 750 751 752 753 754 755
756 757 758 759 760 761 762 763 764 765 766 767
768 769 770 771 772 773 774 775 776 777 778 779
780 781 108 782 784 785 786 787 788 789 790 791
792 445 793 795 796 797 710 752 452 798 802 803
804 805 806 807 808 809 810 811 812 813 814 815
816 817 818 752 819 821 822 823 824 235 825 827
828 829 830 483 831 318 833 835 836 837 838 839
840 841 842 843 844 845 846 847 848 839 849 851
852 853 854 855 856 857 858 859 860 861 862 863
864 865 866 867 868 869 870 871 872 873 874 875
876 877 878 879 880 881 882 883 884 885 886 887
888 889 890 891 892 893 894 895 896 897 898 899
900 901 902 903 904 795 905 907 908 519 909 911
912 913 914 915 916 917 918 919 920 921 922 923
924 884 925 927 928 929 930 931 932 933 934 935
936 937 938 939 930 940 942 943 944 935 945 947
948 839 949 951 952 953 954 955 956 957 958 959
960 961 962 963 964 965 966 967 968 28 969 971
972 973 974 975 976 889 977 979 980 981 982 983
896 984 986 987 988 989 990 991 992 993 994 995
996 997 998 889

43	69	4	683	6	66	8	149	710	657	21	24
12	89	617	17	268	19	800	44	28	23	783	13
26	228	130	29	820	31	241	89	266	35	305	37
38	36	130	15	4	345	129	191	89	4	130	378
50	428	94	89	54	471	56	515	445	67	42	61
560	710	839	65	606	5	346	69	653	12	896	73
686	15	4	78	93	535	10	33	889	15	84	555
86	577	89	87	88	10	8	89	597	386	93	98
519	108	100	101	102	103	104	105	106	107	109	96
111	93	112	113	114	115	116	117	118	119	120	122
4	123	124	125	126	127	128	131	1	175	133	4
134	135	136	137	138	140	182	142	133	143	144	145
146	147	148	151	7	130	152	153	154	155	156	157
159	208	160	163	157	228	164	165	166	167	168	169
172	315	5	173	174	176	89	179	89	29	180	184
208	555	11	185	186	187	188	189	190	192	20	193
194	195	196	199	10	603	200	202	235	203	204	205
206	207	208	209	210	211	212	213	217	5	89	110
218	220	452	221	222	223	224	225	227	483	229	25
231	133	232	235	130	235	236	238	3	240	157	242
30	243	244	245	247	89	248	249	251	150	252	253
254	255	256	257	258	259	262	89	3	263	265	519
267	80	269	16	270	271	272	273	274	275	277	256
278	279	280	281	282	283	284	285	287	3	288	289
290	291	292	293	294	295	297	285	298	299	300	301
302	303	306	346	34	307	308	310	411	312	29	313
314	315	316	317	318	321	1	130	322	323	325	445
326	327	328	329	330	332	292	333	334	335	337	263
338	339	340	341	342	343	344	347	59	2	348	349
350	351	352	353	354	355	357	889	358	359	361	89
362	364	256	365	366	367	368	369	371	208	372	373
375	89	376	377	378	379	380	381	382	384	285	385
387	51	388	389	391	126	392	393	395	21	397	13
398	399	400	402	208	404	1	405	406	407	408	409
410	411	412	413	415	89	416	417	418	419	420	422
89	423	424	425	426	427	430	49	5	431	432	434
29	436	529	437	438	439	440	441	442	443	444	445
446	447	448	450	935	451	452	454	418	455	456	458
13	460	288	461	462	463	464	465	466	468	89	469
472	4	53	473	474	475	476	477	479	480	378	481
483	263	484	485	486	487	488	490	13	491	492	494
29	495	496	498	603	499	500	501	502	503	504	505
506	507	508	509	510	511	512	514	480	516	55	517
518	519	520	521	522	523	524	525	526	528	62	530
3	532	669	533	534	536	76	537	538	539	540	542
89	543	545	47	546	547	548	549	550	552	710	553
554	557	83	6	558	559	561	60	562	564	5	565
566	567	568	569	570	571	572	573	574	575	578	81
85	579	580	581	582	583	584	585	586	587	589	529
590	592	7	593	595	14	596	598	95	600	110	601
602	603	605	669	607	64	609	89	610	611	612	613
614	615	616	618	40	619	620	621	622	623	624	625
626	627	628	629	630	631	632	633	634	635	636	637
639	5	640	641	642	643	644	645	646	647	648	649
650	651	652	654	68	655	656	658	90	659	660	661
662	663	664	665	666	667	668	669	670	672	385	673
674	675	676	677	678	679	680	681	682	684	3	685
687	72	688	689	690	691	693	89	694	695	696	697
698	699	700	701	702	703	704	705	706	707	708	709
711	182	712	713	715	9	716	718	208	719	720	721
722	723	724	725	726	727	728	729	730	731	732	733
734	735	736	738	88	739	740	741	742	743	744	745
746	747	748	749	750	751	752	753	754	755	756	757
758	759	760	761	762	763	764	765	766	767	768	769
770	771	772	773	774	775	776	777	778	779	780	781
782	784	22	785	786	787	788	789	790	791	792	793

795	114	796	797	798	802	89	18	487	803	804	805
806	807	808	809	810	811	812	813	814	815	816	817
818	819	821	28	822	823	824	825	827	322	828	829
830	831	833	752	835	40	836	837	838	839	840	841
842	843	844	845	846	847	848	849	851	884	852	853
854	855	856	857	858	859	860	861	862	863	864	865
866	867	868	869	870	871	872	873	874	875	876	877
878	879	880	881	882	883	884	885	886	887	888	889
890	891	892	893	894	895	896	897	898	899	900	901
902	903	904	905	907	15	908	909	911	62	912	913
914	915	916	917	918	919	920	921	922	923	924	925
927	930	928	929	930	931	932	933	934	935	936	937
938	939	940	942	977	943	944	945	947	3	948	949
951	4	952	953	954	955	956	957	958	959	960	961
962	963	964	965	966	967	968	969	971	710	972	973
974	975	976	977	979	89	980	981	982	983	984	986
89	987	988	989	990	991	992	993	994	995	996	997
998	999	99	3								
43	2	4	3	6	5	8	7	9	90	11	13
12	14	40	17	16	19	18	44	21	23	22	24
26	25	110	29	28	31	30	32	80	35	34	717
38	37	191	15	41	59	1	20	45	46	276	52
50	49	94	324	54	53	56	55	394	58	42	61
60	737	311	65	64	66	459	69	68	433	370	73
72	139	158	78	77	76	493	33	82	178	84	83
86	85	89	88	87	10	241	92	95	51	93	98
97	96	0	0	0	0	0	0	0	0	0	108
0	27	0	0	0	0	0	0	0	0	0	0
121	0	0	0	0	0	0	0	129	130	0	132
0	0	0	0	0	0	74	0	141	0	0	0
0	0	0	0	149	150	0	0	0	0	0	0
0	75	0	0	161	162	0	0	0	0	0	0
0	170	171	0	0	0	175	0	177	81	0	0
181	182	183	0	0	0	0	0	0	0	39	0
0	0	0	0	197	198	0	0	201	0	0	0
0	0	0	0	0	0	0	0	0	214	215	216
0	0	219	0	0	0	0	0	0	226	0	228
0	230	0	0	233	234	0	0	237	0	239	0
91	0	0	0	0	246	0	0	0	250	0	0
0	0	0	0	0	0	0	260	261	0	0	264
0	266	0	268	0	0	0	0	0	0	0	47
0	0	0	0	0	0	0	0	0	286	0	0
0	0	0	0	0	0	0	296	0	0	0	0
0	0	0	304	305	0	0	0	309	0	63	0
0	0	0	0	0	0	319	320	0	0	0	48
0	0	0	0	0	0	331	0	0	0	0	336
0	0	0	0	0	0	0	0	345	346	0	0
0	0	0	0	0	0	0	356	0	0	0	360
0	0	363	0	0	0	0	0	0	71	0	0
0	374	0	0	0	0	0	0	0	0	383	0
0	386	0	0	0	390	0	0	0	57	0	396
0	0	0	0	401	0	403	0	0	0	0	0
0	0	0	0	0	414	0	0	0	0	0	0
421	0	0	0	0	0	0	428	429	0	0	0
70	0	435	0	0	0	0	0	0	0	0	0
0	0	0	0	449	0	0	0	453	0	0	0
457	0	67	0	0	0	0	0	0	0	467	0
0	470	471	0	0	0	0	0	0	0	479	0
0	482	0	0	0	0	0	0	489	0	0	0
79	0	0	0	497	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	513	0	515	0
0	0	0	0	0	0	0	0	0	0	527	0
529	0	531	0	0	0	535	0	0	0	0	0
541	0	0	544	0	0	0	0	0	0	551	0
0	0	555	556	0	0	0	560	0	0	563	0
0	0	0	0	0	0	0	0	0	0	0	576

577	0	0	0	0	0	0	0	0	0	0	588
0	0	591	0	0	594	0	0	597	0	599	0
0	0	0	604	0	606	0	608	0	0	0	0
0	0	0	0	617	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	638	0	0	0	0	0	0	0	0	0	0
0	0	0	0	653	0	0	0	657	0	0	0
0	0	0	0	0	0	0	0	0	0	671	0
0	0	0	0	0	0	0	0	0	0	683	0
0	686	0	0	0	0	0	692	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	710	0	0	0	714	0	0	36	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	62	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	783	0	0	0	0	0	0	0	0	0
0	794	0	0	0	0	799	800	801	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	820	0	0	0	0	0	826	0	0
0	0	0	832	0	834	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	850	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	906	0	0	0	910	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	926	0	0	0	0	0	0	0	0	0	0
0	0	0	0	941	0	0	0	0	946	0	0
0	950	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	970	0	0
0	0	0	0	0	978	0	0	0	0	0	0
985	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1000								
96	76	99	97	99	97	96	96	76	97	76	97
99	76	97	99	97	96	96	99	76	99	97	236
99	97	72	96	96	99	97	76	97	99	97	232
99	97	72	99	77	97	96	97	76	77	72	72
99	97	99	72	99	97	99	97	72	76	99	99
97	72	72	99	97	236	72	99	97	73	72	99
97	73	73	96	76	96	72	99	72	73	96	96
96	96	96	236	96	99	73	76	96	97	96	96
76	96	0	0	0	0	0	0	0	0	0	236
0	76	0	0	0	0	0	0	0	0	0	0
77	0	0	0	0	0	0	0	236	76	0	77
0	0	0	0	0	0	76	0	76	0	0	0
0	0	0	0	236	76	0	0	0	0	0	0
0	76	0	0	76	76	0	0	0	0	0	0
0	76	77	0	0	0	76	0	76	76	0	0
76	76	77	0	0	0	0	0	0	0	236	0
0	0	0	0	77	76	0	0	76	0	0	0
0	0	0	0	0	0	0	0	0	77	76	76
0	0	76	0	0	0	0	0	0	76	0	236
0	76	0	0	76	76	0	0	77	0	76	0
236	0	0	0	0	76	0	0	0	76	0	0
0	0	0	0	0	0	0	76	77	0	0	76
0	236	0	236	0	0	0	0	0	0	0	76
0	0	0	0	0	0	0	0	0	77	0	0
0	0	0	0	0	0	0	76	0	0	0	0
0	0	0	76	236	0	0	0	76	0	76	0
0	0	0	0	0	0	77	76	0	0	0	76
0	0	0	0	0	0	76	0	0	0	0	76
0	0	0	0	0	0	0	0	236	77	0	0
0	0	0	0	0	0	0	76	0	0	0	76

0	0	76	0	0	0	0	0	0	76	0	0
0	76	0	0	0	0	0	0	0	0	76	0
0	236	0	0	0	76	0	0	0	76	0	77
0	0	0	0	76	0	77	0	0	0	0	0
0	0	0	0	0	76	0	0	0	0	0	0
76	0	0	0	0	0	0	236	77	0	0	0
76	0	76	0	0	0	0	0	0	0	0	0
0	0	0	0	76	0	0	0	76	0	0	0
77	0	76	0	0	0	0	0	0	0	76	0
0	77	236	0	0	0	0	0	0	0	76	0
0	76	0	0	0	0	0	0	77	0	0	0
76	0	0	0	76	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	76	0	236	0
0	0	0	0	0	0	0	0	0	0	76	0
77	0	76	0	0	0	236	0	0	0	0	0
76	0	0	76	0	0	0	0	0	0	76	0
0	0	236	77	0	0	0	236	0	0	77	0
0	0	0	0	0	0	0	0	0	0	0	76
236	0	0	0	0	0	0	0	0	0	0	76
0	0	77	0	0	77	0	0	236	0	76	0
0	0	0	76	0	236	0	76	0	0	0	0
0	0	0	0	236	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	77	0	0	0	0	0	0	0	0	0	0
0	0	0	0	236	0	0	0	236	0	0	0
0	0	0	0	0	0	0	0	0	0	76	0
0	0	0	0	0	0	0	0	0	0	236	0
0	236	0	0	0	0	0	76	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	76	0	0	0	77	0	0	76	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	76	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	236	0	0	0	0	0	0	0	0	0
0	76	0	0	0	0	76	236	76	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	236	0	0	0	0	0	76	0	0
0	0	0	76	0	76	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	76	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	77	0	0	0	76	0	0
0	0	0	0	0	0	0	0	0	0	0	0
0	76	0	0	0	0	0	0	0	0	0	0
0	0	0	0	76	0	0	0	0	76	0	0
0	77	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	76	0	0
0	0	0	0	0	76	0	0	0	0	0	0
76	0	0	0	0	0	0	0	0	0	0	0
0	0	0	77								
0											
8	391	982	349	942	516	553	591	630	670	711	753
796	840	885	931	978	42	11	48	68	89	111	675
154	642	137	593	108	566	19	29	228	483		

Program SESLIST

```
      COMPILER(FLD=ABS)
      INTEGER TYPE,CLASS,BLKTYP(100),BLKSIZ(100)
      DIMENSION ISUBLT(3,200),INTFAC(500),IARGS(200)
      EQUIVALENCE (BLKTYP(1),IARGS(1)),(BLKSIZ(1),IARGS(101))
      NLIST=0
      IPTR=1
    1 READ(5,5,END=40) NAME,NARGS,TYPE,CLASS,ISIZE
    5 FORMAT(A6,1X,3I2,1X,I6)
      IF(ISIZE .LT. 1) ISIZE=0
      NLIST=NLIST+1
      IVAR=0
      ISUBLT(1,NLIST)=NAME
      IF(NARGS .NE. -1) GO TO 7
      NARGS=1
      IVAR=1
    7 ITEMP=ISIZE
      FLD(0,6,ITEMP)=NARGS
      FLD(6,4,ITEMP)=CLASS
      FLD(10,3,ITEMP)=TYPE
      ISUBLT(2,NLIST)=ITEMP
      ISUBLT(3,NLIST)=0
      IF(NARGS .EQ. 0) GO TO 1
      FLD(13,1,ISUBLT(2,NLIST))=IVAR
      ISUBLT(3,NLIST)=IPTR
      IF(CLASS .EQ. 7) GO TO 25
      JPTR=IPTR
      INC=1+(NARGS-1)/4
      IPTR=IPTR+INC
      NDPTR=IPTR-1
      NPARAM=3*NARGS
      READ(5,10) (IARGS(I),I=1,NPARAM)
   10 FORMAT(20(3I1,1X))
      KOUNT=0
      DO 20 I=JPTR,NDPTR
      INTFAC(I)=0
      IBIT=-3
      DO 20 K=1,12
      KOUNT=KOUNT+1
      IF(KOUNT .GT. NPARAM) GO TO 1
      IWIDTH=3
      IBIT=IBIT+3
      IF(MOD(KOUNT,3) .EQ. 0) IWIDTH=2
   20 FLD(IBIT,IWIDTH,INTFAC(I))=IARGS(KOUNT)
      GO TO 1
   25 JPTR=IPTR
      INC=1+(NARGS-1)/2
      IPTR=IPTR+INC
      NDPTR=IPTR-1
      READ(5,27) (BLKSIZ(I),BLKTYP(I),I=1,NARGS)
   27 FORMAT(10(I6,1X,I1))
      KOUNT=0
      DO 30 I=JPTR,NDPTR
      INTFAC(I)=0
      IBIT=-3
      DO 30 K=1,2
      KOUNT=KOUNT+1
      IF(KOUNT .GT. NARGS) GO TO 1
      IBIT=IBIT+3
      FLD(IBIT,15,INTFAC(I))=BLKSIZ(KOUNT)
      IBIT=IBIT+15
   30 FLD(IBIT,3,INTFAC(I))=BLKTYP(KOUNT)
      GO TO 1
   40 WRITE(4) NLIST,NDPTR
      WRITE(4) ((ISUBLT(I,J),I=1,3),J=1,NLIST)
      WRITE(4) (INTFAC(I),I=1,NDPTR)
      WRITE(6,50) NLIST
   50 FORMAT(///42X,37H NEW LIST HAS BEEN CREATED CONTAINING,I4,
     $ 6H NAMES)
      STOP
      END
```

Basic Interface Definition File

```
ABS      1 1 4
102
AINT     1 1 4
102
ALOG     1 1 4
102
ALOG10   1 1 4
102
AMAX0   -1 1 4
402
AMAX1   -1 1 4
102
AMIN0   -1 1 4
402
AMIN1   -1 1 4
102
AMOD     2 1 4
102 102
AIMAG    1 1 4
202
ATAN     1 1 4
102
ATAN2    2 1 4
102 102
CABS     1 1 4
202
CCOS     1 2 4
202
CEXP     1 2 4
202
CLOG     1 2 4
202
CMPLX    2 2 4
102 102
CONJG    1 2 4
202
COS      1 1 4
102
CSIN     1 2 4
202
CSQRT    1 2 4
202
DABS     1 3 4
302
DATAN    1 3 4
302
DATAN2   2 3 4
302 302
DBLE     1 3 4
102
DCOS     1 3 4
302
DEXP     1 3 4
302
DIM      2 1 4
102 102
DLOG     1 3 4
302
DLOG10   1 3 4
302
DMAX1   -1 3 4
302
DMIN1   -1 3 4
302
DMOD     2 3 4
302 302
DSIGN    2 3 4
302 302
DSIN     1 3 4
302
DSQRT    1 3 4
302
EXP      1 1 4
102
FLOAT    1 1 4
402
IABS     1 4 4
402
IDIM     2 4 4
402 402
IDINT    1 4 4
302
IFIX     1 4 4
102
INT      1 4 4
102
ISIGN    2 4 4
402 402
MAX0    -1 4 4
402
MAX1    -1 4 4
102
MIN0    -1 4 4
402
MIN1    -1 4 4
102
MOD      2 4 4
402 402
REAL     1 1 4
202
SESCOM   2 0 7      25
    13 0     12 4
SIGN     2 1 4
102 102
SIN      1 1 4
102
SNGL     1 1 4
302
SQRT     1 1 4
102
TAN      1 1 4
102
TANH     1 1 4
102
```

INITIAL DISTRIBUTION

Copies

10	NAVSEA PMS304-32 White
2	NAVSEA PMS405-40 Cuthbert
12	DDC

CENTER DISTRIBUTION

1	18/1809
1	1802.2 Frenkiel
1	1802.4 Theilheimer
1	1809.3 D. Harris (Central Depository, CMLD)
1	182 Camara
1	1826 Culpepper
30	1826 Wybraniec
1	184 Lugt
1	185 Corin
1	186 Sulit
1	189 Gray
1	1890 Taylor
30	5214.1 Reports Distribution
1	522

Microfiche copies

30	1826 Wybraniec

DTNSRDC ISSUES THREE TYPES OF REPORTS

(1) DTNSRDC REPORTS, A FORMAL SERIES PUBLISHING INFORMATION OF PERMANENT TECHNICAL VALUE, DESIGNATED BY A SERIAL REPORT NUMBER.

(2) DEPARTMENTAL REPORTS, A SEMIFORMAL SERIES, RECORDING INFORMATION OF A PRELIMINARY OR TEMPORARY NATURE, OR OF LIMITED INTEREST OR SIGNIFICANCE, CARRYING A DEPARTMENTAL ALPHANUMERIC IDENTIFICATION.

(3) TECHNICAL MEMORANDA, AN INFORMAL SERIES, USUALLY INTERNAL WORKING PAPERS OR DIRECT REPORTS TO SPONSORS, NUMBERED AS TM SERIES REPORTS; NOT FOR GENERAL DISTRIBUTION.


DATE
ILME